Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos

Error estimation and adaptivity in deformable solid dynamics

Doctoral Thesis

Luis María Lacoma Aller

Ingeniero Industrial

Director: Ignacio Romero Olleros

Doctor Ingeniero Industrial

Madrid, May 2006

Departamento de Mecánica de Medios Continuos y
Teoría de Estructuras

Escuela Técnica Superior de Ingenieros de Caminos,
Canales y Puertos

Universidad Politécnica de Madrid

Error estimation and adaptivity in deformable solid dynamics

Doctoral Thesis

Luis María Lacoma Aller

Ingeniero Industrial

Director: Ignacio Romero Olleros
Doctor Ingeniero Industrial

Madrid, May 2006

Error estimation and adaptivity in deformable solid dynamics

Doctoral Thesis
Universidad Politécnica de Madrid

Madrid, May 2006

This text was typeset using LaTeX

Luis María Lacoma Aller
Ingeniero Industrial

Advisor: Ignacio Romero Olleros
Doctor Ingeniero Industrial

Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos
Departamento de Mecánica de Medios Continuos y Teoría de Estructuras

Universidad Politécnica de Madrid
Profesor Aranguren s/n
Madrid 28040

Teléfono: (+34) 91 336 5232
Fax: (+34) 91 336 6702
Correo electrónico: `ll@mecanica.upm.es`
Página web: `http://w3.mecanica.upm.es/~ll`

Abstract

In this dissertation we study the error due to the temporal discretization in numerical simulations of deformable solid dynamics .

For that, we start from the semidiscrete equations of motion and analyze the conditions that an estimator must satisfy to compute accurate results. With them, two typical error estimators are analyzed.

Later, we propose a novel methodology for the formulation of *a posteriori* error estimators for the most common time-stepping methods employed in solid and structural dynamics. The estimators obtained by means of this methodology are accurate even in non-smmoth problems, they can be applied both in linear and non-linear problems and can be easily implemented in finite element codes. The proposed methodology is applied to construct error estimators for Newmark's method and for the HHT method. The good performance of these new estimators is investigated in several numerical simulations.

The information given by these estimators is the starting point for developing adaptive algorithms. They change automatically the time step size to ensure that the error is smaller than a tolerance, but keeping the computational cost as low as possible. The algorithms presented are based on control theory, and different techniques applied in this field are employed to optimize the adaptive strategies. The final adaptive schemes are very simple formulas which are easily implemented in existing codes.

Three non-linear numerical examples are presented to validate the good behaviour of the adaptive algorithms described.

Keywords: time integration; error estimation; solid dynamics; adaptivity; control theory

Acknowledgements

En primer lugar quiero agradecer al profesor Ignacio Romero por su magnífica dirección durante estos años y cuyo fruto es este trabajo. Supongo que el haber sido su primer doctorando no habrá sido fácil para él, pero desde estas líneas quiero expresar mi más sincera gratitud por su guía. Su disponibilidad, el gran tiempo que hemos pasado juntos y sus siempre acertados comentarios, han hecho posibile llevar a buen término este trabajo.

También deseo agradecer al profesor José Mª Goicolea por haberme permitido venir e incorporarme a su grupo. No puedo olvidarme tampoco de los demás componentes del Grupo de Mecánica Computacional, Juan José Arribas, Felipe Gabaldón, Juan Carlos García y Santiago Muelas y Yolanda Cabrero, por el buen ambiente de trabajo creado.

Evidentemente ocupan un lugar especial los compañeros de doctorado que forman o han formado parte del grupo, Francisco Calvo, Francisco Riquelme, Roberto Ortega, Adelardo González, Daniel Iglesias, Claudio García, Beatriz Sanz, Damon Afkari así como los demás compañeros de la Escuela: Alvaro, Rafael, Miguel, Carola con los que he compartido muchos momentos de trabajo y también de diversión. Sin duda, entre ellos destaca el doctor Javier Rodríguez del que tanto he aprendido.

He de agradecer a la Consejería de Educación de la Comunidad de Madrid por haberme concedido una beca F.P.I. y proporcionarme financiación durante estos años.

Por último, a mis padres y hermana debo agradecerles el ayudarme a venir a Madrid y el apoyo que me siguen dando a pesar de la distancia, y a Manuela por hacerme pasar momentos tan felices a su lado.

To my parents,
and my sister

To Manuela

"Everything should be as simple as possible, but not simpler"
Albert Einstein

*"El primer paso de la ignorancia es presumir de saber,
y muchos sabrían si no pensasen que saben"*
Baltasar Gracián

Contents

★ Parts of this thesis have appeared in the following publications:

- I. Romero & L. M. Lacoma [2006]. "Error estimation for the semidiscrete equations of solid and structural mechanics". *Computer Methods in Applied Mechanics and Engineering* **195**: pp. 2674-2696.

- I. Romero & L. M. Lacoma [2006]. "A methodology for the formulation of error estimators for time integration in solid and structural dynamics". *International Journal for Numerical Methods in Engineering* **66**: pp. 635-660.

- L. M. Lacoma & I. Romero [2006]. "Estimación de error en dinámica de sólidos deformables". *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* **22**: pp. 45-62.

- I. Romero & L. M. Lacoma [2006]. "Analysis and improved methods for the error estimation of numerical solutions in solid and structural dynamics". *Advances in Computational Multibody Dynamics*. Springer-Verlag. In press.

- L. M. Lacoma & I. Romero [2006]. "Error estimation for the HHT method in non-linear solid dynamics". Submitted to *Computers & Structures*.

Chapter 1

Introduction: motivation and scope

The objective of the research presented in this dissertation is to design efficient algorithms which allow to solve a solid dynamics problem with the desired accuracy keeping the computational cost as low as possible.

This initial chapter is an introduction to the work that will be developed throughout the next chapters and tries to motivate the rest of the investigation. Also, a short review of the state of the art is given. We conclude the chapter with an overview of the topics presented in this dissertation.

1.1 Motivation

Nowadays, computational methods are becoming a more and more demanding tool for solving different types of problems in all brands of engineering. Their use allows to reduce costs in the product development and in the design process, since they avoid or reduce the use of experimental prototypes which entail both high economical and time costs.

However, any numerical method is only capable of computing an approximated solution to the problem. Therefore, this numerical solution contains errors that come from different sources. The first source of errors is due to represent mathematically a physical reality. That is, the process of describing a physical phenomenon by means of a model with equations entails simplifications and/or approximations. The discipline that studies these errors is called “model validation” (see, e.g. Ladevèze et al. [2006a] and Ladevèze et al. [2006b] in the context of structural dynamics). In this work, this type of errors will not be taken into account.

Once a mathematical model is built, the equations which describe it must be solved. Only in simple models an analytical solution can be obtained. This fact leads, in many cases, to the use of numerical methods and computers for their solution. No numerical method is capable to compute the exact solution to the problem but it calculates an approximation to it. This is the second source of errors which include errors inherent to the numerical method (discretization errors, computer round-off errors), etc. This class of errors cannot be known exactly but, at least, they can be computed approximately through the use of error estimators.

From the information provided by an error estimator, any analyst has an objective tool to determine whether a simulation has been carried out with the sufficient accuracy and therefore, to know the reliability of the results obtained.

Besides providing information about the accuracy of a simulation, the use of error estimators allows the development of adaptive schemes. In general, the characteristic of an adaptive algorithm is that they solve any simulation with a prescribed tolerance keeping the computational cost as small as possible.

In this dissertation, we are going to concentrate in problems of solid or structural dynamics. Many applications of engineering require to know the dynamic response of structures or deformable solids. Some examples of such applications are vehicle dynamics, design of buildings, bridges, dams (or any other structure) under earthquakes or wind loads, impacts between different bodies (e.g. bird impact on aircraft structures), design and analysis of mechanisms, etc.

The equations that model these processes can be very complex and they only have analytical solutions in simple cases. Therefore, most of the problems must be solved using numerical methods introducing therefore, errors in the solution.

Usually, a problem in computational solid dynamics is solved in two stages. This process is known as semidiscretization and it will be explained in Chapter 2. Basically, this method discretizes spatially the continuous problem by the finite element method reaching an initial value problem of ordinary differential equations. This is the starting point of this work. This system of equations is solved using a time-stepping method and we are interested in studying the error made due to the temporal discretization. All this process is depicted in Figure 1.1.

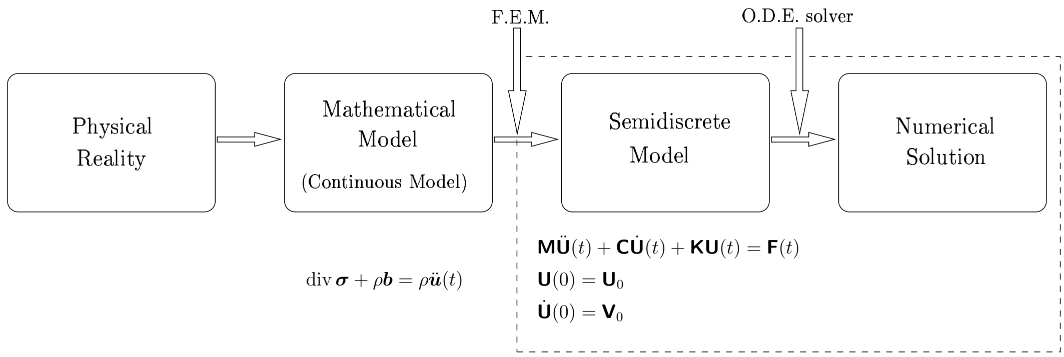


Figure 1.1: Stages for solving a problem of solid or structural dynamics.

The part of the full solution which is studied in this dissertation is boxed with a dashed line in Figure 1.1. Starting with the semidiscrete equations (where the time is still a continuous variable) these are solved by means of time-stepping methods. This type of methods computes the solution step by step, i.e., known the solution of the up to an instant, they compute the solution at the next instant, advancing in time. Our final objective is to select automatically the time step size in such way that the estimated error is equal to a given tolerance.

1.1.1 A motivating example

The better way to understand the importance of using an error estimator in a solid dynamics problem is through a numerical example. The following simulation, although is an academic problem is very illustrative.

The problem consists of the study of a deformable double pendulum, each composed of a spring and a mass. Both springs have stiffness $k = 1000$ N/m and the masses are of value $m = 1$ kg. Initially, both springs are in horizontal position and the pendulum falls under the action of gravity, being the gravity constant a tenth of the terrestrial gravity.

Different simulations of this system are performed during 50 s and using two numerical methods which will be referred to as method 1 and method 2. These methods will be discussed in detail in the Chapter 2.

First, we use the method 1 with two different time step sizes: $\Delta t = 0.5$ s and $\Delta t = 0.05$ s. Figure 1.2 plots the angle (measured in degrees) of the first spring. We can observe that during the first 10 s the response obtained with both time steps is very similar. However, from this instant up to the end, the evolution of the pendulum in both cases differs dramatically.

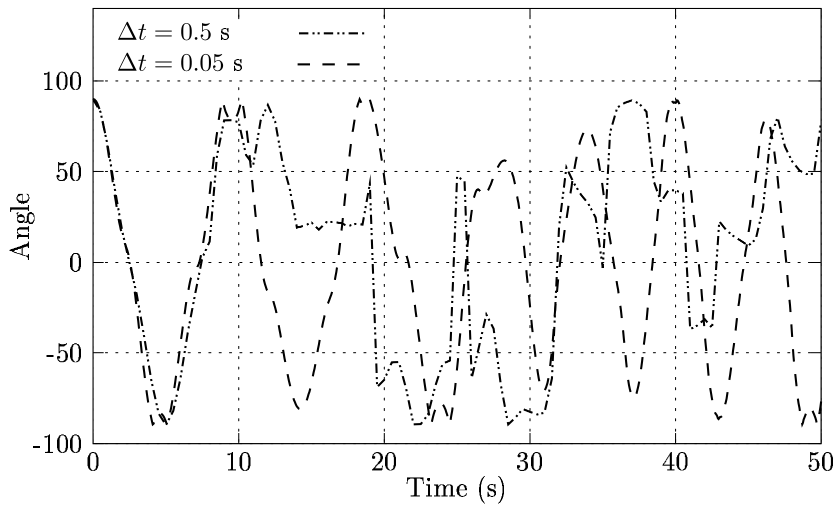


Figure 1.2: Angle (measured in degrees) of the first spring respect to the vertical. Analysis carried out using the method 1.

Now, we perform the simulation with a time step $\Delta t = 0.05$ s but using the method 1 and two variants of the method 2. As before, Figure 1.3 depicts the angle of the first spring for the three methods compared. Up to $t = 33$ s all methods compute the same response, but from this point the results start to differ.

These differences in the evolution of the pendulum can also be observed in Figure 1.4.

After analyzing these simulations, a main question arises: which of the simulations performed is more accurate?. It seems logical to think that the smaller the time step is, the more accurate the numerical solution becomes. However, if the problem is solved using different methods it is difficult to know a priori which of the results is better.

Throughout this work we will try to answer this question. For that, algorithms that allow to estimate the error made in a dynamic simulation will be developed.

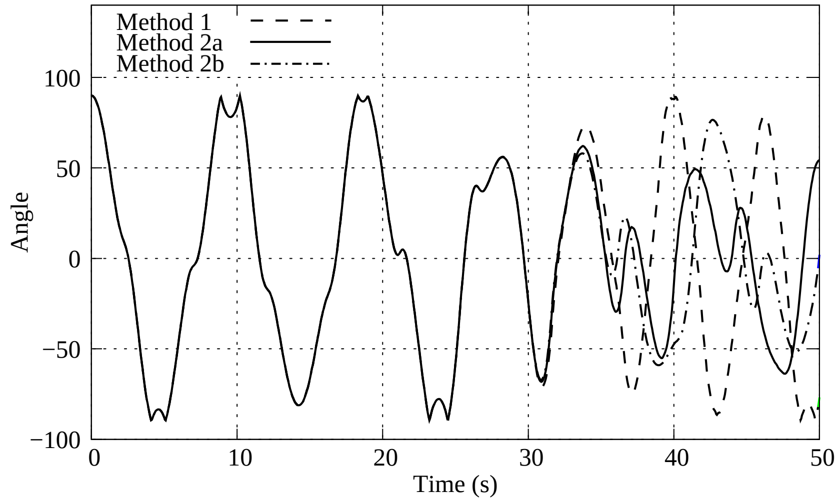


Figure 1.3: Angle (measured in degrees) of the first spring respect to the vertical. Analysis carried out with a constant time step $\Delta t = 0.05$ s using the method 1 and two variants of the method 2.

1.2 Goals

This works focus on the solution in time of the semidiscrete equations which arise in solid dynamics problems in an efficient and accurate way.

The goal of the work described in this dissertation is twofold. On the one hand, the first task is to develop accurate and reliable error estimators for solid and structural dynamics problems. On the other hand, the second task is to design efficient adaptive techniques of the time step size to reduce so the computational cost of the dynamic analysis. To achieve both goals the following subtasks are performed:

i. The statement of the necessary conditions that an error estimator must fulfil to provide correct results.

ii. The analysis of existing estimators for solid dynamics problems. From this analysis we can detect their drawbacks and their range of applicability.

iii. The design of new estimators which improve the accuracy of the existing ones keeping a small computational cost.

iv. The development of efficient adaptive in time schemes. They allow to carry out the dynamic analyses with the desired accuracy reducing the computational effort.

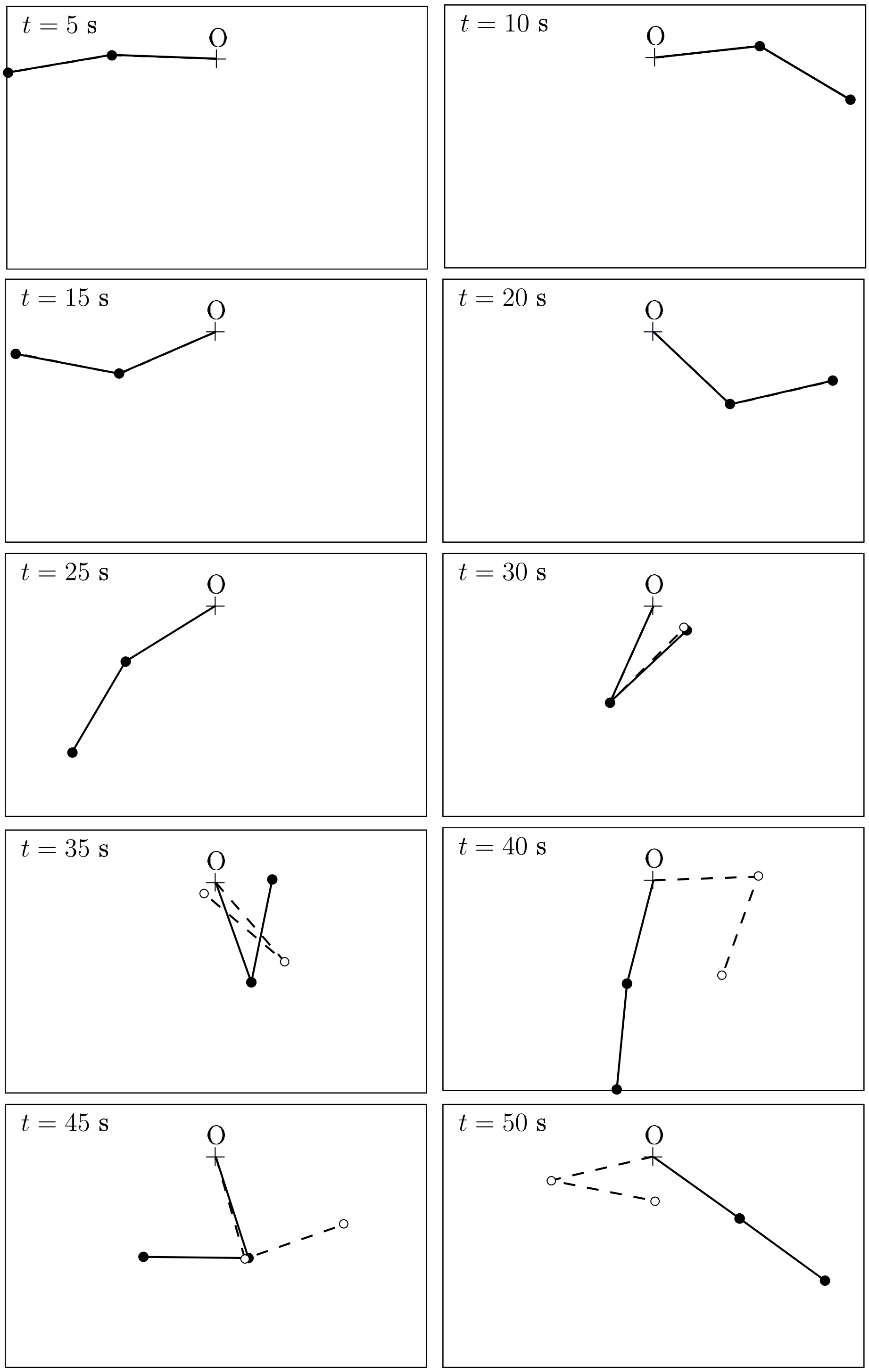


Figure 1.4: Evolution of the double pendulum. In continuous line: results obtained with the method 2 and $\Delta t = 0.05$ s. In dashed line: results obtained with the method 1 and $\Delta t = 0.05$ s. Snapshots shown every 5 seconds starting at $t = 5$ s.

1.3 Background

The field of error estimation in computational mechanics has undergone an important growth over the last years, becoming nowadays in a very active field. This results in, among others, the existence of international congresses such as the "International Conference on Adaptive Modeling and Simulation (ADMOS)" which has celebrated two editions in 2003 and 2005 and the next edition will be held in 2007. Likewise, the journal *Computers Methods in Applied Mechanics in Engineering* has dedicated a special issue to this topic*. Besides great amount of research articles published, in the last years it has even appeared monographs and books dealing with this theme (see e.g. the books by AINSWORTH & ODEN [2000] and by AKIN [2005]).

However, many of the advances in the field of error estimation and adaptivity have been developed for static problems being scant the works for the dynamic case. A possible reason for this discrepancy can be the different nature of the governing equations that model each case. Whereas in the static case the equations are of elliptic type, in the dynamic one the governing equation is hyperbolic. This fact causes that the methods developed for estimating the error in static problems cannot be applied to dynamic analysis.

1.3.1 Error estimation in time in solid dynamics problems

As commented before, this work focuses on the study of the error made when the semidiscrete equations of motion are solved by means of a time-stepping method. That is, we are only going to study the error due to the temporal discretization. A dynamic simulation is an evolution in time problem and thus, a time-stepping method computes the solution step by step. Therefore, for computing the solution at an instant, the solution at previous instants must be known.

Two of the most used commercial finite element softwares in industry are ANSYS and ABAQUS/Standard. For dynamic simulations, ANSYS has implemented Newmark's method and ABAQUS/Standard uses the HHT method. The analysis of the accuracy of the solution in both programs is based on the concept of "half-step residual" (HIBBIT ET AL. [2004], Section 2.4.1.). It consists of knowing the solution at two consecutive time instants, computing the residual at the middle of the interval. This value provides a qualitative information about

*CMAME Volume 195, Issues 4-6, Pages 205-480.

the accuracy of the solution but it does not compute an objective estimation of the error made.

The first work about error estimation in structural dynamics problems was carried out by ZIENKIEWICZ & XIE [1991]. They developed an estimator in displacements for Newmark's method. Following an alternative procedure ZENG ET AL. [1992] obtained this same estimator and CHOI & CHUNG [1996] designed one for the θ-Wilson method. In WIBERG & LI [1993] a more complete estimator is built for Newmark's method (this one takes into account the contribution to the error both of the displacements and velocities). Both the estimators of ZIENKIEWICZ & XIE [1991] and WIBERG & LI [1993] will be analyzed in detail in Chapter 3.

GÈRADIN & CARDONA [1997] extended the procedure presented in ZIENKIEWICZ & XIE [1991] to the HHT method and HULBERT & JANG [1995] to the generalized-α method. All these works develop *a posteriori* error estimators. That is, it is needed to compute the numerical solution at an instant to know the error at that time. In contrast CHUNG ET AL. [2003] built an *a priori* estimator, i.e., it is capable of giving an estimation of the error at an instant without calculating the solution there, using information of the previous steps.

All these estimators are built following the same idea. They compute an "improved" solution (a more accurate solution than the numerical one) by means of expansions in Taylor's series of the solution about the initial time of the step. Each estimator uses a different approach to approximate the derivatives higher than second.

As will be seen in Chapter 3, all of them have some shortcomings in different situations. For instance, the estimator of ZIENKIEWICZ & XIE [1991] (and all those derived from it) neglects an important part of the error, causing that the error estimate is not an upper bound for the true error. On the other hand, the estimator of WIBERG & LI [1993] exhibits a good performance in smooth problems, whereas it provides unrealistic large estimates in non-smooth simulations.

To analyze the performance of an error estimator, ROMERO & LACOMA [2006b] have recently introduced the conditions that any estimator must satisfy to provide accurate error bounds. Also, in that same work, the estimator of WIBERG & LI [1993] is studied detecting those cases where this method overestimate in a large magnitude the error. These results will be discussed in Chapter 3.

In Romero & Lacoma [2006c] a new strategy for designing error estimators for linear dynamic problems is proposed. The authors particularize it for the case of using Newmark's method. This strategy is based on building the improved solution by means of a Gauss-Lobatto quadrature and the resulting error estimator exhibits a better performance than the existing ones. This novel strategy will be explained in Chapter 4.

In Romero & Lacoma [2006a] and Lacoma & Romero [2006a] that strategy is extended to non-linear problems and it is applied to develop an error estimator for Newmark's method in the former work and for HHT method in the latter one.

1.3.2 Adaptive methods in deformable solid dynamics

As commented before, a solid or structural dynamics simulation is a evolution in time problem. That is, the numerical method starts at an initial time and advances in time. The interval between an instant and the next one is called time step and its size is denoted by Δt. The larger the time step size is, the smaller the number of steps are and thus, the smaller is the computational effort. However, we are interested in solving the analysis with a prescribed tolerance. From this point of view, we are interested in advancing with small time steps to provide accurate results.

The fulfilment of this double requirement (efficiency and accuracy) can be achieved using adaptive schemes. Any adaptive algorithm computes an estimation for the error and compared it with a tolerance. According to this comparison the algorithm increases or decreases the time step size.

Next, this idea is explained graphically. Suppose that a step is computed and the error estimate is smaller than the tolerance. Then, this step is accepted and the analysis is continued with a new time step size (see Figure 1.5). Suppose now the opposite, that is, that the error estimate is greater than the tolerance. In this situation, the step is rejected and it must be recomputed with a smaller time step size (see Figure 1.6).

There exist several strategies for increasing or decreasing the time step size. Most of them are based on more or less heuristic rules. Such rules have been used for years in many softwares for solving ODEs and, nowadays they follow being implemented in, e.g., the suite Sundials (Hindmarsh et al. [2005]) (a general purpose solver for nonlinear and differential/algebraic equations) or the finite element program Abaqus/Standard.

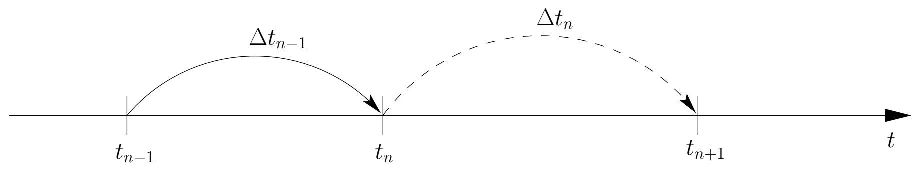


Figure 1.5: Accepted step. The analysis advances to the next step.

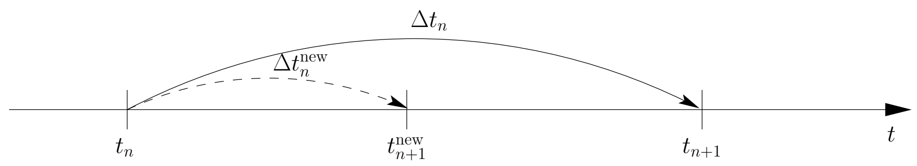


Figure 1.6: Rejected step. The step is rewind and computed again with a smaller time step.

These strategies often require defining the value of several parameters which are adjusted through numerical experiments and they have little or no solid mathematical basis.

Many of the works that propose error estimators for deformable solid simulations also gives adaptive algorithms. As an example, the strategy followed in Zienkiewicz & Xie [1991] is a follows. To increase the time step, the estimated error has to be smaller than a percentage of the tolerance (usually a half). If the error is larger than this quantity but smaller than the tolerance, the time step is held constant. The formula used for increasing and decreasing the time step is the same (it will be referred to as standard or I-controller and it will be described in Chapter 5, Section 5.1). In Zeng et al. [1992], Choi & Chung [1996], Li & Wiberg [1998] and Chung et al. [2003] the adaptive strategy is the same than in Zienkiewicz & Xie [1991], whereas in Hulbert & Jang [1995] a very similar strategy with slight modifications is proposed.

In Gèradin & Cardona [1997] and Noels et al. [2003] the range of possibilities for increasing the time step size is widened. The zone under the tolerance is divided in several intervals and in each of them the update formula used is different. We refer to the original papers for a complete description of each strategy.

In this work, we investigate an approach to this subject based on control theory. The first attempt of studying the problem of adaptation of the time step from this point of view was carried out in Gustafsson et al. [1988]. In it, the authors showed that the standard time step algorithm was seen, from this new approach, as an I (integral) controller. It is well established in control theory that there exist other controllers more efficient, such as the

PI (proportional-integral) and the PID (proportional-integral-derivative). Following this novel line, K. GUSTAFFSON studies the behaviour of PI-controllers for explicit (GUSTAFSSON [1991]) and implicit Runge-Kutta methods (GUSTAFSSON [1994]). Later, all these concepts are studied in depth in SÖDERLIND [2002] where the behaviour of the different controllers is analyzed in the frequency domain. In SÖDERLIND [2003], the same author introduces the PID controllers for selecting the time step size and in SÖDERLIND & WANG [2006] studies how the different controllers affect to the stability of ODE codes. Some of these advances are summarized in the recent work by SÖDERLIND [2006].

All works described above are developed in a general context of ODEs. With respect to more specific fields, VALLI ET AL. [2002] apply strategies of time step selection based on a PID controller for the simulation of couple viscous flow and heat transfer. However, to the author's knowledge the use of this type of techniques have not been explored in the field of deformable solid dynamics.

1.4 Overview

This dissertation is divided in six chapters, in which the first one is this introduction. An outline of the remainder of the work is given next.

Chapter 2 introduces the initial value problem of differential equations which govern a problem of structural or deformable solid dynamics. Also, two numerical methods for solving these type of equations are discussed.

Chapter 3 deals with error estimation in solid and structural dynamics problems. For that, the concepts of global error and local error are defined. Likewise, an important relation between them is given. From these results, we address the conditions which an error estimator must fulfil to provide accurate estimates. These conditions are used to analyze two typical estimators existing in the literature. From this analysis some shortcomings of these estimators are deduced. The identification of these shortcomings are the point of departure to develop a new error estimator, whose performance is studied through several numerical simulations.

In Chapter 4 we introduce a new strategy for designing error estimators for the most common time-stepping methods. We applied it to develop estimators for Newmark's method an for the HHT method. Both of them can be applied to both linear and non-linear problems. Their behavior and good performance are analyzed in several numerical simulations.

Chapter 5 deals with adaptive techniques in time from the point of view of control theory. That is, how the time step size is selected automatically along the analysis in such a way that the estimated error keeps close to a prescribed tolerance. The strategies presented are compared in three complex non-linear simulations.

Finally, Chapter 6 concludes this dissertation by summarizing the main results and by pointing out the future lines of research.

Chapter 2

The initial value problem of deformable solid dynamics

The purpose of this chapter is to describe and to state the problem of solid and structural dynamics, introducing for that the equations which govern it. In Section 2.1 these equations are presented. In Section 2.2, we analyze this type of problem and several interesting results are deduced. These results will be employed in the following chapters. Also, the concept of energy norm is introduced. Finally, Section 2.3 presents two numerical methods widely employed in solid and structural dynamic simulations for solving these problems.

2.1 Semidiscrete equations of motion. Problem definition

A problem of deformable solid dynamics can be described, from a mathematical point of view, by two forms. The first one consists of a set of non-linear partial differential equations and it is referred to as the "strong" or classical form of the problem. These equations establish the equilibrium at each point of the solid. The second alternative is to express the problem by means of a set of integral equations. It is called the "weak" or variational form and studies the equilibrium of the solid in a global way. The "weak" form is the starting point for developing numerical schemes of the type of the finite element method.

Both the strong formulation and the weak formulation are equivalent in the sense that one can be deduced from the other, and vice versa. The reader can consult the books by BRAESS [1997] or by HUGHES [2000] for a proof of this equivalence.

From the weak formulation of the solid dynamics problem, two strategies are followed in the literature for solving numerically these equations. The first strategy consists of approximating both the spatial variables and the time using the finite element method. This strategy is called discontinuous space-time Galerkin method (HULBERT [1989]) and it establishes a natural framework in which develop error estimates (see, e.g, LI & WIBERG [1998], SCHLEUPEN & RAMM [2000] and ROMERO & LACOMA [2004]). However, the drawback of this method is its high computational cost.

The second strategy handles the spatial variables and the temporal variable in a different way. The process is as follows. First, the spatial part is discretized using generally the finite element method (although another type of method such as a mesh-free can be used instead of the finite element method). This discretization leads to a system of second-order ordinary differential equations (ODEs), where the time is still a continuous variable. This system of ODEs can now be solved by means of one of many existing methods; for example, RUNGE-KUTTA methods, ADAMS method, BDF methods, etc. However, in the last fifty years, several numerical methods have been specifically designed for this type of differential equations. Two of the most representative ones will be discussed in Section 2.3. This second strategy receives the name of *semidiscretization* and in the mathematical literature it is referred to as *method of lines*. This technique is addressed in detail in HUGHES [2000] in the context of linear elastodynamics and in BELYTSCHKO ET AL. [2000] for non-linear dynamic problems.

The drawback of the semidiscretization strategy is that it makes more difficult a global analysis of the whole problem. However, this approach leads to developing simple and computationally efficient numerical methods.

Likewise, this same type of system of ODEs also arises in other fields such as multibody systems (see e.g. the monograph by GARCÍA DE JALÓN & BAYO [1993]).

This work focuses on the study the error due to the temporal discretization, i.e. the error made in the stage of solving the resulting system of ODEs. Thus, our starting point is the following initial value problem (IVP), which is defined in the time interval $\mathcal{I} = [0, T]$:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{U}}(t) + \mathbf{F}_{\text{int}}(\mathbf{U}(t), \dot{\mathbf{U}}(t)) = \mathbf{F}(t, \mathbf{U}(t)) \ , & t \in \mathcal{I} \ , \\ \mathbf{U}(0) = \mathbf{U}_o \ , \\ \dot{\mathbf{U}}(0) = \mathbf{V}_o \ , \end{cases} \tag{2.1}$$

where $\mathbf{U}(t)$ is the vector of nodal displacements which is the unknown of the problem, $\dot{\mathbf{U}}(t)$ the vector of velocities and $\ddot{\mathbf{U}}(t)$ the vector of accelerations, all with dimension n_{dofs}, being n_{dofs} the number of degrees of freedom of the model. The matrix $\mathbf{M}$ is the mass matrix, which is symmetric and positive definite and it has dimensions $n_{\text{dofs}} \times n_{\text{dofs}}$. $\mathbf{F}_{\text{int}}(\mathbf{U}(t), \dot{\mathbf{U}}(t))$ is the vector of internal forces and, in a general case, it can depend non linearly on displacements and velocities. The vector $\mathbf{F}(t, \mathbf{U}(t))$ is the vector of external forces and it can be a function of the time and also of the displacements. Finally, $\mathbf{U}_o$ and $\mathbf{V}_o$ are, respectively, the initial displacements and velocities.

Remark 2.1. In a linear problem, the vector of internal forces $\mathbf{F}_{\text{int}}$ is a linear combination of displacements and velocities and the vector of external forces only depends on the time, i.e.,

$$\begin{aligned} &\mathbf{F}_{\text{int}}(\mathbf{U}(t), \dot{\mathbf{U}}(t)) = \mathbf{C}\dot{\mathbf{U}}(t) + \mathbf{K}\mathbf{U}(t) \ , \\ &\mathbf{F}(t, \mathbf{U}(t)) = \mathbf{F}(t) \ , \end{aligned} \tag{2.2}$$

where the matrices $\mathbf{C}$ and $\mathbf{K}$ are the damping and stiffness matrices, respectively, and they are symmetric and positive semidefinite. Both of them has dimensions $n_{\text{dofs}} \times n_{\text{dofs}}$. With the definitions indicated above, a linear problem is stated as:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{U}}(t) + \mathbf{C}\dot{\mathbf{U}}(t) + \mathbf{K}\mathbf{U}(t) = \mathbf{F}(t) \ , & t \in [0, T] \ , \\ \mathbf{U}(0) = \mathbf{U}_o \ , \\ \dot{\mathbf{U}}(0) = \mathbf{V}_o \ . \end{cases} \tag{2.3}$$

□

In a real problem, the initial value problem (2.1) can be highly non-linear. The vector of internal forces can include both geometric and material nonlinearities. Geometric nonlinearities group large displacements (and rotations in structural elements), large deformations, contact forces between different bodies, etc. Within the material nonlinearities are included complex behavior of materials, such as, phenomenon of hiperelasticity, plasticity, viscoplasticity and damage, among others. Regarding the vector of external forces, it can contain some non-linear term as in the case of existing follower loads.

2.2 Analysis of the IVP of solid dynamics

Although the scope of this work is the non-linear dynamics, a more complete theoretical analysis can be carried out in the linear case. Hence, in this section, the analysis is restricted to the case where the internal and external forces are defined as in (2.2).

Problem defined in (2.3) is a second-order system of ordinary differential equations. For the analysis which will be carried out, a transformation to a first-order system is useful. The resulting first-order system has a dimension which is twice the dimension of the original problem (2.2). For this purpose, a new variable denoted by $\boldsymbol{z}(t)$ is defined. This variable is of dimension $2n_{\mathrm{dof}}$ and collects the displacement and velocity vectors. Likewise, the initial conditions are collecting into $\boldsymbol{z}_o$. Thus,

$$\boldsymbol{z}(t) = \left\{ \begin{array}{c} \mathbf{U}(t) \\ \dot{\mathbf{U}}(t) \end{array} \right\} , \quad \boldsymbol{z}_o = \left\{ \begin{array}{c} \mathbf{U}_o \\ \mathbf{V}_o \end{array} \right\} . \tag{2.4}$$

Furthermore, a constant matrix denoted by $\mathbf{B}$ and a vector $\mathbf{g}(t)$, of dimensions $2n_{\mathrm{dof}} \times 2n_{\mathrm{dof}}$ and $2n_{\mathrm{dof}}$ respectively, are introduced, such that:

$$\mathbf{B} = \left[\begin{array}{cc} \mathbf{0} & \mathbf{1} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{array} \right] , \quad \mathbf{g}(t) = \left\{ \begin{array}{c} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F}(t) \end{array} \right\} . \tag{2.5}$$

With the new variables defined in (2.4) and (2.5), the initial value problem (2.2) is written in a more compact form as:

$$\left\{ \begin{array}{ll} \dot{\boldsymbol{z}}(t) = \mathbf{B}\boldsymbol{z}(t) + \mathbf{g}(t) , & t \in \mathcal{I} , \\ \boldsymbol{z}(0) = \boldsymbol{z}_o . & \end{array} \right. \tag{2.6}$$

Remark 2.2. A necessary condition for the uniqueness of solution of differential equation (2.6) is the continuity of the function $\mathbf{g}(t)$. It means that the external force $\mathbf{F}(t)$ must be a continuous function. In next chapter, in Section 3.6, we discuss the case when discontinuities exist. □

Since the matrix $\mathbf{B}$ is constant, the initial value problem (2.6) has the following closed form solution (see, for example, ASCHER & PETZOLD [1998], Chapter 2),

$$\boldsymbol{z}(t) = \mathbf{H}(t)\boldsymbol{z}_o + \mathbf{h}(0,t) , \tag{2.7}$$

where

$$\mathbf{H}(t) = \mathrm{e}^{\mathbf{B}t} \quad \text{and} \quad \mathbf{h}(0,t) = \int_0^t \mathbf{H}(t-\tau)\mathbf{g}(\tau)\,\mathrm{d}\tau \ . \tag{2.8}$$

Remarks 2.3.

i. The factor $\mathbf{H}(t)\boldsymbol{z}_o$ is the solution of the homogeneous equation associated to (2.6) and will be denoted by $\boldsymbol{z}_h(t)$. In other words, $\boldsymbol{z}_h(t)$ is the solution of problem (2.6) when the external forces are zero.

$$\boldsymbol{z}_h(t) = \mathbf{H}(t)\boldsymbol{z}_o \Longrightarrow \boldsymbol{z}(t) = \boldsymbol{z}_h(t) + \mathbf{h}(0,t) \ . \tag{2.9}$$

ii. The vector of external forces $\mathbf{F}(t)$ has only influence on the exact solution through the integral $\mathbf{h}(0,t)$ defined in $(2.8)_2$ which is the particular solution of problem (2.6) □

Throughout this dissertation, the size of different vectors will have to be compared. For that, the concept of norm of a vector is introduced next. There exists many different norms. In this work we will use the *energy norm* which is the "natural" norm in dynamic problems, since it takes into account both the contribution of displacements and velocities. In some works about error estimation in dynamic problems, such as in Chung et al. [2003]; Gèradin & Cardona [1997]; Zeng et al. [1992]; Zienkiewicz & Xie [1991], the authors use the L_2-norm for measuring the error and solution vectors of the problem. This norm takes only into account the displacement variables neglecting the contribution of velocities.

The *energy norm* is very useful for studying the convergence and stability of solutions in elastodynamic problems (see Romero [2002] and Romero [2004]). In the following definition the concept of *energy norm* is introduced.

Definition 2.1. *Let $\boldsymbol{z}(t) = \langle \mathbf{U}(t), \dot{\mathbf{U}}(t)\rangle$ be the solution of the initial value problem (2.3). The* energy norm *of $\boldsymbol{z}(t)$, denoted by $||\boldsymbol{z}(t)||_E$, is a function which maps vectors of dimension $2n_{\mathrm{dof}}$ into a non-negative real number and it is defined as:*

$$\begin{aligned} ||\cdot||_E : \mathbb{R}^{2n_{\mathrm{dof}}} &\longrightarrow [0,\infty) \\ \boldsymbol{z}(t) &\longmapsto ||\boldsymbol{z}(t)||_E := \left[\frac{1}{2}\boldsymbol{z}(t)\cdot\begin{bmatrix}\mathbf{K} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}\boldsymbol{z}(t)\right]^{1/2} \ . \end{aligned} \tag{2.10}$$

It can be easily proved that function $||\cdot||_E$ indeed fulfils all properties of a norm.

Expanding the matrix product in (2.10), the energy norm reads:

$$||\boldsymbol{z}(t)||_E^2 = \underbrace{\frac{1}{2}\dot{\mathbf{U}}(t)\cdot\mathbf{M}\dot{\mathbf{U}}(t)}_{\substack{\text{KINETIC}\\\text{ENERGY}}} + \underbrace{\frac{1}{2}\mathbf{U}(t)\cdot\mathbf{K}\mathbf{U}(t)}_{\substack{\text{POTENTIAL}\\\text{ENERGY}}} \ . \tag{2.11}$$

That is, the square of the energy norm is the sum of the kinetic energy of the system and the deformation potential energy. Thus, the unit of energy norm, in the international system, is $\sqrt{J}$ where J denote the unit of energy, i.e., the Joule.

Remarks 2.4.

i. The energy norm has been defined for vector $\boldsymbol{z}(t)$. However, it can be used to measure any vector of dimension $2n_{\text{dof}}$ with the correct units, such us, a numerical solution or different vectors of errors which will be introduced in the next chapter.

ii. The energy norm (2.10) has been defined for a linear problem. In a nonlinear problem, the stiffness matrix is not constant since it depends on the displacements. In this work, in order to compute the energy norm of some vector when a nonlinear problem is being solved, the initial stiffness matrix is selected, i.e.,

$$||\boldsymbol{z}(t)||_E^2 = \frac{1}{2}\dot{\mathbf{U}}(t)\cdot\mathbf{M}\dot{\mathbf{U}}(t) + \frac{1}{2}\mathbf{U}(t)\cdot\mathbf{K}_o\mathbf{U}(t) \ , \tag{2.12}$$

where

$$\mathbf{K}_o = \left.\frac{\partial\mathbf{F}_{\text{int}}(\mathbf{U},\mathbf{V})}{\partial\mathbf{U}}\right|_{(\mathbf{U},\mathbf{V})=(\mathbf{U}_o,\mathbf{V}_o)} \ . \tag{2.13}$$

□

In Chapter 3, bounds for errors in dynamic problems will be obtained. These will be based on a bound of the matrix $\mathbf{H}(t)$, which was defined in $(2.8)_1$. Next, we present two propositions where a bound for this matrix is investigated.

Proposition 2.1. *In absence of external forces the energy of the exact solution of the problem (2.6) does not increase in time.*

PROOF: To prove this proposition, it suffices to check the following expression

$$\frac{\mathrm{d}}{\mathrm{d}t}||\boldsymbol{z}(t)||_E^2 \leqslant 0 \ . \tag{2.14}$$

From definition of the energy norm (2.10), and rewritten it as in (2.11), it follows

$$||\boldsymbol{z}(t)||_E^2 = \frac{1}{2}\mathbf{U}(t) \cdot \mathbf{K}\mathbf{U}(t) + \frac{1}{2}\dot{\mathbf{U}}(t) \cdot \mathbf{M}\dot{\mathbf{U}}(t) \ . \tag{2.15}$$

By derivating the expression (2.15) with respect to time, it can be deduced that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}||\boldsymbol{z}(t)||_E^2 &= \frac{1}{2}\dot{\mathbf{U}}(t) \cdot \mathbf{K}\mathbf{U}(t) + \frac{1}{2}\mathbf{U}(t) \cdot \mathbf{K}\dot{\mathbf{U}}(t) + \\ &\quad \frac{1}{2}\ddot{\mathbf{U}}(t) \cdot \mathbf{M}\dot{\mathbf{U}}(t) + \frac{1}{2}\dot{\mathbf{U}}(t) \cdot \mathbf{M}\ddot{\mathbf{U}}(t) \\ &= \mathbf{U}(t) \cdot \mathbf{K}\dot{\mathbf{U}}(t) + \dot{\mathbf{U}}(t) \cdot \mathbf{M}\ddot{\mathbf{U}}(t) \\ &= \mathbf{U}(t) \cdot \mathbf{K}\dot{\mathbf{U}}(t) + \dot{\mathbf{U}}(t) \cdot [-\mathbf{C}\dot{\mathbf{U}}(t) - \mathbf{K}\mathbf{U}(t)] \\ &= -\dot{\mathbf{U}}(t) \cdot \mathbf{C}\dot{\mathbf{U}}(t) \\ &\leqslant 0 \ .\end{aligned} \tag{2.16}$$

The second equality results from symmetry of the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$, and the last inequality in (2.16) is a consequence of the positive semidefiniteness of the damping matrix $\mathbf{C}$. ■

Remarks 2.5.

i. In systems without external forces and without damping ($\mathbf{F}(t) = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$), the energy of the system remains constant. This fact can be easily deduced from (2.16).

ii. When external forces are zero, the integral $\mathbf{h}(0, t)$ vanishes and thus, in expression (2.16) the solution $\boldsymbol{z}(t)$ can be changed by $\boldsymbol{z}_h(t)$, obtaining

$$\frac{\mathrm{d}}{\mathrm{d}t}||\boldsymbol{z}_h(t)||_E \leqslant 0 \ . \tag{2.17}$$

This result will be used in the following proposition. □

Proposition 2.2. *The energy norm of the matrix* $\mathbf{H}(t)$ *defined in* $(2.8)_1$ *is bounded by 1, i.e.,*

$$||\mathbf{H}(t)||_E \leqslant 1 \ . \tag{2.18}$$

PROOF: Before proving this result, we must define the concept of energy norm of a matrix. Let $\boldsymbol{x}$ be a vector of dimension $2n_{\mathrm{dof}}$ and $\mathbf{D}$ a matrix of dimensions

$2n_{\mathrm{dof}} \times 2n_{\mathrm{dof}}$. Then, the energy norm of the matrix $\mathbf{D}$ is defined as follows:

$$||\mathbf{D}||_E := \max_{||\boldsymbol{x}||_E=1} ||\mathbf{D}\boldsymbol{x}||_E = \max_{\boldsymbol{x}\neq\mathbf{0}} \frac{||\mathbf{D}\boldsymbol{x}||_E}{||\boldsymbol{x}||_E} \ . \tag{2.19}$$

Now, applying the expression of energy norm of a matrix (2.19) to the matrix $\mathbf{H}(t)$, it is concluded that,

$$||\mathbf{H}(t)||_E = \max_{\boldsymbol{z}_o\neq\mathbf{0}} \frac{||\mathbf{H}(t)\boldsymbol{z}_o||_E}{||\boldsymbol{z}_o||_E} = \max_{\boldsymbol{z}_o\neq\mathbf{0}} \frac{||\boldsymbol{z}_h(t)||_E}{||\boldsymbol{z}_o||_E} \leqslant 1 \ , \tag{2.20}$$

where the Proposition 2.1 and the Remark 2.5.ii have been used in the last inequality. ■

2.3 Numerical methods for solving the semidiscrete equations of solid dynamics

As explained before, there exists many numerical methods for solving system of differential equations. A rigorous analysis and description of the different existing methods is beyond the scope of this work. In this section, we concentrate in two methods specifically designed to handle and to take advantage of the structure of second-order ODEs.

A classification of the numerical methods for solving the equations of dynamics can be to distinguish between *explicit* and *implicit* methods. *Explicit* methods compute directly the solution at t_{n+1} from known values at previous times. In contrast, *implicit* methods also take into account values at t_{n+1} and therefore, they lead to the resolution of a system of linear equations at each time step.

Explicit methods employ time steps much smaller that implicit ones being them restricted under stability conditions. For this reason, they are recommended to be used in problems where a small scale of time is needed in order to characterize properly the physical phenomenon under study. This type of problems include impacts, contacts at high velocities, etc. In contrast, implicit methods allow to advance with larger time steps and they are preferred in structural applications dominated by low-frequency response.

In this work, we will concentrate in the error estimation for implicit methods, and more specifically a new estimator will be developed in Chapter 4 for Newmark's method and for the HHT method. Both methods are described in

Sections 2.3.2 and 2.3.3, respectively, and they can be possibly considered as two of the most widely used time-stepping methods in the context of structural and solid dynamics.

Other implicit methods are: Houbolt's method (HOUBOLT [1950]), θ-Wilson method (WILSON [1968]), and the generalized-α method (CHUNG & HULBERT [1993]). All these methods are discussed and analyzed, among others, in BELYTSCHKO [1983] and in the works by T.J.R. HUGHES (HUGHES [2000] and HUGHES [1983]). A wide compilation and classification on time-stepping methods in the context of solid dynamics can be found in TAMMA ET AL. [2000].

2.3.1 Preliminary concepts on time-stepping methods

Before describing the two time-stepping methods used in this work, we review some theoretical aspects on time-stepping methods.

For this, we start introducing the notation $\mathcal{O}(\cdot)$, which is continually used throughout this dissertation. This symbol, sometimes called "big-O", is one of the Landau symbols and it is defined as follows. Given a function $f(t)$, then $f(\Delta t)$ is said to be of size $\mathcal{O}(\Delta t^k)$ if and only if

$$|f(\Delta t)| < c \cdot \Delta t^k \ , \tag{2.21}$$

with c being a positive real constant independent of Δt.

There are three concepts which define the behavior of a time-stepping methods for ODEs. They are the concepts of convergence, consistency and stability which are defined next.

Definition 2.2. *Let $\boldsymbol{z}(t_n)$ be the exact solution to the initial value problem (2.6) and $\tilde{\boldsymbol{z}}_n$ be a numerical approximation to it. Then, a time-stepping method is said to be* convergent *if the numerical solution tends to the exact solution when the time step size goes to zero, i.e.,*

$$\lim_{\Delta t \to 0} ||\boldsymbol{z}(t_n) - \tilde{\boldsymbol{z}}_n|| = 0 \ . \tag{2.22}$$

Definition 2.3. *(*GEAR *[1971]) Let the* local truncation error *be the amount by which the solution of the differential equation fails to satisfy the equation used in the numerical method. A time-stepping method is* accurate *or* consistent *with order p if the local truncation error is of size $\mathcal{O}(\Delta t^{p+1})$ (p is the order of accuracy or the rate of convergence of the algorithm).*

Definition 2.4. *A time-stepping method is said to be* stable *if there exists a time step size τ such that a change in the starting conditions produces a bounded change in the numerical solution. This has to be verified for all time step Δt such that*

$$0 < \Delta t < \tau \ . \tag{2.23}$$

When $\tau = \infty$ the time-stepping method is said to be *unconditionally stable*, that is, the method is stable for any value of the time step selected. Otherwise, the method is *conditionally stable.*

These three properties presented herein are discussed in detail in the classical texts of numerical analysis of ODEs. Therefore, we refer to the reader to the monographs of Hairer et al. [1987], Gear [1971], Lambert [1991] or Stuart & Humphries [1996] for a deeper study of these topics.

2.3.2 Newmark family of methods

Newmark's method (Newmark [1959]) is, in fact, a family of time-stepping schemes, each with different properties. Some of them are widely employed in structural dynamic problems.

Newmark's method is formulated as follows. Firstly, the time interval $\mathcal{I} = [0, T]$ is split in N subintervals $\mathcal{I}_i = [t_i, t_{i+1}]$, with $0 \leqslant i \leqslant N-1$. Given a set of numerical values for displacements, velocities and accelerations at instant t_n, denoted respectively, $\mathbf{U}_n$, $\mathbf{V}_n$ and $\mathbf{A}_n$, the algorithmic values at the following instant of time t_{n+1} are computed solving the following algebraic system of equations:

$$\begin{cases} \mathbf{M}\mathbf{A}_{n+1} + \mathbf{F}_{\text{int}}(\mathbf{U}_{n+1}, \mathbf{V}_{n+1}) = \mathbf{F}(t_{n+1}, \mathbf{U}_{n+1}) \ , \\ \mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t_n \mathbf{V}_n + \dfrac{\Delta t_n^2}{2}[(1-2\beta)\mathbf{A}_n + 2\beta\mathbf{A}_{n+1}] \ , \\ \mathbf{V}_{n+1} = \mathbf{V}_n + \Delta t_n[(1-\gamma)\mathbf{A}_n + \gamma\mathbf{A}_{n+1}] \ , \end{cases} \tag{2.24}$$

where $\Delta t_n = t_{n+1} - t_n$. The initial displacements and velocities ($\mathbf{U}_o$ and $\mathbf{V}_o$) are the initial conditions of the problem and therefore, they are known. The initial acceleration is computed from the balance equation, i.e.,

$$\mathbf{A}_0 = \mathbf{M}^{-1}\left[\mathbf{F}(t_o, \mathbf{U}_o) - \mathbf{F}_{\text{int}}(\mathbf{U}_o, \mathbf{V}_o)\right] \ . \tag{2.25}$$

Changing the values of β and γ results in a wide range of methods. For example, if $\gamma = 0.5$ the resulting methods are second-order accurate. Otherwise, they are only of first-order. The combination $\gamma = 0.5$ and $\beta = 0.25$ produces the well-known *trapezoidal rule* or *average acceleration method*, which is the most commonly used within Newmark family and methods with $\beta = \dfrac{(0.5+\gamma)^2}{4}$ and $\gamma > 0.5$ have unconditionally stability. In Table 2.1 a summary of these methods is presented.

$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	**Name**	**Type**	**Order**
0.25	0.5	Trapezoidal rule	Implicit	2
0	0.5	Central differences	Explicit	2
$(0.5+\gamma)^2/4$	$0.5 < \gamma \leqslant 1$	Dissipative	Implicit	1

Table 2.1: Some of the methods of the Newmark family and their properties

From Table 2.1, we observe that to build a method with numerical dissipation in high-frequencies, then the parameter γ has to be greater than 0.5 (Hughes [2000]), losing thus one order of accuracy. The HHT method was developed to overcome this drawback.

2.3.3 The Hilber-Hughes-Taylor (HHT) method

The Hilber-Hughes-Taylor method (Hilber et al. [1977]), henceforth referred to as the HHT method, is a one parameter multi-step implicit method specifically designed to solve second-order differential equations. It can be seen as a generalization of Newmark's method and its main property is that it allow to introduce numerical damping of higher frequencies without degrading the second-order accuracy of the method.

The HHT method follows a formulation similar to Newmark's method. In first place the time interval $[0,T]$ is split in N consecutive subintervals $[t_i, t_{i+1}]$. Given a set of numerical values for displacements, velocities and accelerations at instant t_n, the algorithmic values at the following instant of time are computed solving the following algebraic system of equations:

$$\begin{cases} \mathbf{M}\mathbf{A}_{n+1} + \mathbf{F}_{\text{int}}(\mathbf{U}_{n+\alpha}, \mathbf{V}_{n+\alpha}) = \mathbf{F}(t_{n+\alpha}, \mathbf{U}_{n+\alpha}) \ , \\ \mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t_n \mathbf{V}_n + \dfrac{\Delta t_n^2}{2}[(1-2\beta)\mathbf{A}_n + 2\beta\mathbf{A}_{n+1}] \ , \\ \mathbf{V}_{n+1} = \mathbf{V}_n + \Delta t_n[(1-\gamma)\mathbf{A}_n + \gamma\mathbf{A}_{n+1}] \ , \end{cases} \tag{2.26}$$

where the initial acceleration is computed via expression (2.25).

Note that approximations for algorithmic displacements and velocities $\mathbf{U}_{n+1}$ and $\mathbf{V}_{n+1}$ are the same as in Newmark's method. Variables denoted by $(\bullet)_{n+\alpha}$ are computed by the convex combination:

$$(\bullet)_{n+\alpha} = (1-\alpha)(\bullet)_n + \alpha(\bullet)_{n+1} \; . \tag{2.27}$$

In the system of equations (2.26), three different parameters appear, which are α, β and γ. In order to conserve the second-order accuracy and an unconditional stability of the method, the parameters are related as follows:

$$\beta = \left(1 - \frac{\alpha}{2}\right)^2 \; , \qquad \gamma = \frac{3}{2} - \alpha \; , \qquad \alpha \in [0.7, 1] \; . \tag{2.28}$$

That is, given a value for α, then the method is completely defined. This is the reason why, in the literature, this method is also called α-method. Selecting $\alpha = 1$, the trapezoidal rule is recovered and the smaller α is, the more numerical damping is introduced to the system.

Chapter 3

Error estimation in dynamic problems

In the previous chapter, the equations which govern a solid dynamic problem were presented. Also, two numerical methods for solving this type of problems were discussed. Both methods are widely employed in commercial finite element softwares. Among others, programs such as ANSYS [2005] and PLAXIS [2005] implement the Newmark scheme, whereas ABAQUS/Standard (HIBBIT ET AL. [2004]) uses the HHT method to simulate dynamic problems.

As any numerical tool, the time-stepping methods described in this work are not capable of computing the exact solution of the problem. Hence, the computed numerical solutions have errors which are not known *"a priori"*. These errors can be reduced in two ways: either by decreasing the time step or by employing a more accurate numerical method. Both alternatives entail a considerable increment in the computational cost of the simulation, since the first one will require more time steps to complete the analysis, whereas the second alternative will generally need more function evaluations and also to store more information. In addition, they do not assess how accurate the results are.

In contrast, the use of *"a posteriori"* error estimates provides important information about the accuracy of the results. Using this information, a designer has an objective tool to decide if a simulation has been carried out with sufficient accuracy and, therefore, to be able to know if the results are reliable.

This chapter deals with error estimation in solid or structural dynamic problems. First, in Section 3.1, we review some fundamental concepts and prove an important bound which will be useful to develop error estimators. In Section 3.2, relations between *"a priori"* bounds and *"a posteriori"* estimates are investigated. Next, conditions which an error estimator must fulfil are discussed in Section 3.3. With these conditions, we analyze in Section 3.4 two existing error estimators identifying their shortcomings and their range of applicability. In Section 3.5, a new simple error estimator is formulated. This estimator overcomes some drawbacks detected in the existing ones. Finally, we conclude the chapter presenting a numerical simulation where the concepts discussed in previous sections are examined.

3.1 *"A priori"* error bounds

Consider the initial value problem of solid dynamics expressed as a first order system of ordinary differential equations given by equation (2.6). Let $\boldsymbol{z}(t)$ be the exact solution of this problem, recalling that this vector collects both displacements and velocities. Let $\tilde{\boldsymbol{z}}_0$, $\tilde{\boldsymbol{z}}_1$, $\tilde{\boldsymbol{z}}_2$, ..., $\tilde{\boldsymbol{z}}_N$ be a numerical solution at successive time instants obtained using a time-stepping method such as the Newmark or the HHT method. Then, the most important error is the global error (also known as total or true error). It is defined as follows: the global error at time t_n, denoted by $\boldsymbol{E}_n$, is the difference between the exact solution and the numerical solution, i.e,

$$\boldsymbol{E}_n := \boldsymbol{z}(t_n) - \tilde{\boldsymbol{z}}_n \ . \tag{3.1}$$

From (3.1), it is clear that to calculate the global error requires knowing the exact solution of the initial value problem. This is only possible in very simple problems and hence, in a real problem this error will never be known exactly.

Henceforth, the analysis presented is restricted to the linear case. The main reason to do this is that in the linear regimen a more complete analysis can be performed.

As will be seen, a simple way to approximate the value of the global error is via the accumulation of local errors. These local errors are only defined in each local initial value problem. This local problem is formulated as follows:

Given the initial conditions $\mathbf{U}_n$ and $\mathbf{V}_n$, find the function $\boldsymbol{u}^{[n]}(t)$ defined in the interval $\mathcal{I}_n$ that verifies

$$\begin{cases} \mathbf{M}\ddot{\boldsymbol{u}}^{[n]}(t) + \mathbf{C}\dot{\boldsymbol{u}}^{[n]}(t) + \mathbf{K}\boldsymbol{u}^{[n]}(t) = \mathbf{F}(t) \ , \qquad t \in \mathcal{I}_n = [t_n, t_{n+1}] \ , \\ \boldsymbol{u}^{[n]}(t_n) = \mathbf{U}_n \ , \\ \dot{\boldsymbol{u}}^{[n]}(t_n) = \mathbf{V}_n \ . \end{cases} \tag{3.2}$$

This local problem (3.2) can also be expressed as a first order system of ordinary differential equations. Denoting by $\boldsymbol{y}^{[n]}(t_n) = \langle \boldsymbol{u}^{[n]}(t_n), \dot{\boldsymbol{u}}^{[n]}(t_n)\rangle$ the exact local solution, this problem could be written as:

$$\begin{cases} \dot{\boldsymbol{y}}^{[n]}(t) = \mathbf{B}\boldsymbol{y}^{[n]}(t) + \mathbf{g}(t) \qquad t \in \mathcal{I}_n \ , \\ \boldsymbol{y}^{[n]}(t_n) = \tilde{\boldsymbol{z}}_n \ , \end{cases} \tag{3.3}$$

where $\tilde{\boldsymbol{z}}_n = \langle \mathbf{U}_n, \mathbf{V}_n \rangle$ is the initial condition and $\mathbf{B}$ and $\mathbf{g}(t)$ were defined in (2.5).

Remarks 3.1.

i. The initial conditions $\mathbf{U}_n$ and $\mathbf{V}_n$ in the local problem (3.2) and (3.3) are the displacements and the velocities obtained by the time-stepping method at the end of $\mathcal{I}_{n-1}$.

ii. The superscript $(\bullet)^{[n]}$ refers to the interval where the local problem is defined. We adopt the convention that $(\bullet)^{[n]}$ indicates the interval $[t_n, t_{n+1}]$, $(\bullet)^{[n+1]}$ the interval $[t_{n+1}, t_{n+2}]$ and so forth.

iii. It is important to emphasize that $\boldsymbol{y}^{[n]}(t_{n+1}) \neq \boldsymbol{y}^{[n+1]}(t_{n+1})$, since the later variable is the initial condition of the local problem defined in $\mathcal{I}_{n+1}$, i.e, $\boldsymbol{y}^{[n+1]}(t_{n+1}) = \langle \mathbf{U}_{n+1}, \mathbf{V}_{n+1}\rangle$. And the former is the exact solution at t_{n+1} of the local problem defined in the interval $\mathcal{I}_n$. □

As explained in Section 2.2, first order systems with the structure of the local problem (3.3) have a closed form solution. This solution is built using the operators $\mathbf{H}(t)$ and $\mathbf{h}(0,t)$ which were defined in (2.8). Thus, the solution of the local problem reads

$$\boldsymbol{y}^{[n]}(t_{n+1}) = \mathbf{H}(\Delta t_n)\tilde{\boldsymbol{z}}_n + \mathbf{h}(t_n, t_{n+1}) \ , \tag{3.4}$$

recalling that $\boldsymbol{y}^{[n]}(t)$ is only defined in the time interval $[t_n, t_{n+1}]$.

Once a local problem has been defined, the concept of local error can be introduced. Let $\boldsymbol{y}^{[n]}(t_{n+1})$ be the exact solution at t_{n+1} of the local problem and $\tilde{\boldsymbol{z}}_{n+1}$ a numerical solution, then the local error $\boldsymbol{e}_{n+1}$ is

$$\boldsymbol{e}_{n+1} := \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \ . \tag{3.5}$$

To compute the local error it is mandatory to know the exact solution of the local problem. Although, in a linear problem, computing this local solution is strictly possible by evaluating exactly the operators $\mathbf{H}(\Delta t_n)$ and $\mathbf{h}(t_n, t_{n+1})$, it would require an extraordinary effort of calculation.

Moreover, suppose that we could solve exactly the first local problem. Then, as depicted in Figure 3.1, the local solution obtained at t_1 is equal to the exact solution $\boldsymbol{z}(t_1)$. Therefore, the initial condition of the second local problem is also $\boldsymbol{z}(t_1) = \tilde{\boldsymbol{z}}_1 = \boldsymbol{y}^{[0]}(t_1) = \boldsymbol{y}^{[1]}(t_1)$. Again, solving exactly this second local problem, we compute the exact solution at t_2, $\boldsymbol{z}(t_2)$, and so on. In conclusion, solving exactly each local problem would lead to the exact solution $\boldsymbol{z}(t)$.

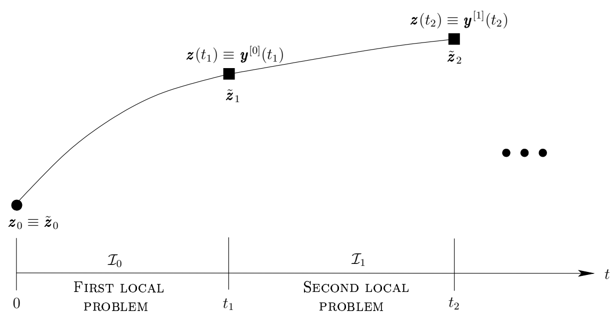


Figure 3.1: Solving exactly each of the local problems.

The usefulness of the local errors is that they allow to build an upper bound for the global error. This result is discussed in the following theorem.

Theorem 3.1. *The energy norm of the exact global error at time t_n is bounded by the sum of exact local errors measured in the energy norm, i.e,*

$$||\boldsymbol{E}_n||_E \leqslant \sum_{i=0}^{n} ||\boldsymbol{e}_i||_E \ . \tag{3.6}$$

Proof: By completeness, two different proofs of this theorem will be shown. The first one assuming a constant time step, and the second one with a variable time step. Both of them follow the same scheme and they are based on the induction principle.

i. FIRST PROOF: Assume that the time step intervals $\mathcal{I}_i$ have all equal lengths, i.e, $\Delta t_o = \Delta t_1 = \Delta t_2 = ... = \Delta t_n = \Delta t$ and let $\mathbf{H} := \mathbf{H}(\Delta t)$. First, let us show that the global error can be expressed as

$$\boldsymbol{E}_n = \sum_{i=1}^{n} \mathbf{H}^{n-i} \boldsymbol{e}_i \ . \tag{3.7}$$

After the first step, at time $t = t_1 = \Delta t$, the exact global error is, using expression (3.1), $\boldsymbol{E}_1 = \boldsymbol{z}(\Delta t) - \tilde{\boldsymbol{z}}_1$. For this same initial time step, the exact local error, defined in (3.5), is $\boldsymbol{e}_1 = \boldsymbol{y}^{[0]}(\Delta t) - \tilde{\boldsymbol{z}}_1$, and as $\boldsymbol{z}(\Delta t) = \boldsymbol{y}^{[0]}(\Delta t)$, the exact global error equals the exact local error verifying (3.7). Then, it only needs to be proven that if the equality (3.7) is satisfied at time t_n, it is also satisfied at instant t_{n+1}.

$$\begin{aligned}
\boldsymbol{E}_{n+1} &= \boldsymbol{z}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \boldsymbol{z}((n+1)\Delta t) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}\boldsymbol{z}(n\Delta t) + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\tilde{\boldsymbol{z}}_n + \boldsymbol{E}_n) + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}\boldsymbol{E}_n + \mathbf{H}\tilde{\boldsymbol{z}}_n + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}\boldsymbol{E}_n + \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}\boldsymbol{E}_n + \boldsymbol{e}_{n+1} \\
&= \sum_{i=1}^{n+1} \mathbf{H}^{n+1-i} \boldsymbol{e}_i \ ,
\end{aligned} \tag{3.8}$$

which proves (3.7). If the norm energy is taken in both members of (3.7), and the triangle inequality and bound (2.18) are used, the error bound (3.6) is easily deduced.

ii. SECOND PROOF: In this second proof the time step does not have to be constant and thus, it can vary along the analysis. Following an identical process than in the first proof, a relation between the exact global errors and exact local errors is deduced.

$$
\begin{aligned}
\boldsymbol{E}_{n+1} &= \boldsymbol{z}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\Delta t_n)\boldsymbol{z}(t_n) + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\Delta t_n)[\tilde{\boldsymbol{z}}_n + \boldsymbol{E}_n] + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\Delta t_n)\boldsymbol{E}_n + \mathbf{H}(\Delta t_n)\tilde{\boldsymbol{z}}_n + \mathbf{h}(t_n, t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\Delta t_n)\boldsymbol{E}_n + \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \mathbf{H}(\Delta t_n)\boldsymbol{E}_n + \boldsymbol{e}_{n+1} \\
&= \sum_{i=1}^{n+1} \left[\prod_{j=1}^{n+1-i} \mathbf{H}(\Delta t_{n+1-j}) \right] \boldsymbol{e}_i \ .
\end{aligned} \tag{3.9}
$$

Finally, taking the energy norm in (3.9) and using the triangle inequality and the stability bound (2.18), the error bound (3.6) is obtained:

$$
\begin{aligned}
||\boldsymbol{E}_{n+1}||_E &= \left\| \sum_{i=1}^{n+1} \left[\prod_{j=1}^{n+1-i} \mathbf{H}(\Delta t_{n+1-j}) \right] \boldsymbol{e}_i \right\|_E \\
&\leqslant \sum_{i=1}^{n+1} \left\| \prod_{j=1}^{n+1-i} \mathbf{H}(\Delta t_{n+1-j}) \right\|_E ||\boldsymbol{e}_i||_E \\
&\leqslant \sum_{i=1}^{n+1} ||\boldsymbol{e}_i||_E \ .
\end{aligned} \tag{3.10}
$$

■

Remark 3.2. The bound established in Theorem 3.1 only holds if the norm used to measure the errors is the energy norm and the problem is linear. □

Note that the result obtained in Theorem 3.1 does not allow to compute an error estimate, since both local errors and global error are unknown. However, it establishes an important relation which will be crucial in the next section.

3.2 *"A posteriori"* error estimates

As mentioned before, it is not possible to compute exactly neither the global error nor the local errors. However, in this section it is shown that if we compute a sufficiently accurate approximation to the exact local solution $\boldsymbol{y}^{[n]}(t_{n+1})$, then an error estimate is easily built.

If we use a p-order accurate method to compute a numerical solution of problem (2.3), it is well known that then the local error is of size $\mathcal{O}(\Delta t^{p+1})$ (see e.g. HAIRER ET AL. [1987]). For instance, for the HHT method, which is second-order accurate, the size of each local error is $\mathcal{O}(\Delta t^3)$.

Now, a new algorithmic solution, denoted by $\tilde{\boldsymbol{y}}^{[n]}_{n+1}$, is introduced. This solution which henceforth will be referred to as the *improved solution* must be, at least, one order more accurate than the solution provided by the numerical method. With help of this *improved solution*, the exact local error can be split in two parts:

$$\begin{aligned} \boldsymbol{e}_{n+1} &= \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\ &= \left(\boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}^{[n]}_{n+1}\right) + \left(\tilde{\boldsymbol{y}}^{[n]}_{n+1} - \tilde{\boldsymbol{z}}_{n+1}\right) \ . \end{aligned} \tag{3.11}$$

The first part in (3.11), i.e. the term $\boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}^{[n]}_{n+1}$, is the difference between the exact local solution and the *improved solution*. It is denoted by $\boldsymbol{\eta}_{n+1}$ and is of size $\mathcal{O}(\Delta t^{p+k+1})$, with $k \geqslant 1$, that is,

$$\boldsymbol{\eta}_{n+1} := \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}^{[n]}_{n+1} = \mathcal{O}(\Delta t^{p+k+1}), \qquad k \geqslant 1 \ . \tag{3.12}$$

The second term in (3.11), denoted by $\boldsymbol{\theta}_{n+1}$, is a *local error estimate* and its size is $\mathcal{O}(\Delta t^{p+1})$, i.e.,

$$\boldsymbol{\theta}_{n+1} := \tilde{\boldsymbol{y}}^{[n]}_{n+1} - \tilde{\boldsymbol{z}}_{n+1} = \mathcal{O}(\Delta t^{p+1}) \ . \tag{3.13}$$

Thus, combining definitions (3.12) and (3.13) the following relation holds

$$\frac{||\boldsymbol{\eta}_{n+1}||_E}{||\boldsymbol{\theta}_{n+1}||_E} = \frac{\mathcal{O}(\Delta t^{p+k+1})}{\mathcal{O}(\Delta t^{p+1})} = \mathcal{O}(\Delta t^k), \quad k \geqslant 1 \ , \tag{3.14}$$

or in other words, the quantity $\boldsymbol{\eta}_{n+1}$ is negligible in comparison to $\boldsymbol{\theta}_{n+1}$.

By inserting the relations (3.12) and (3.13) into the expression of the local error (3.11) and making use of result given in (3.14), the local error reads:

$$
\begin{aligned}
\boldsymbol{e}_{n+1} &= \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{z}}_{n+1} \\
&= \left(\boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}^{[n]}_{n+1}\right) + \left(\tilde{\boldsymbol{y}}^{[n]}_{n+1} - \tilde{\boldsymbol{z}}_{n+1}\right) \\
&= \underbrace{\boldsymbol{\eta}_{n+1}}_{\mathcal{O}(\Delta t^{p+1+k})} + \underbrace{\boldsymbol{\theta}_{n+1}}_{\mathcal{O}(\Delta t^{p+1})} \\
&\approx \boldsymbol{\theta}_{n+1} \ .
\end{aligned}
\tag{3.15}
$$

That is, whenever $\boldsymbol{\eta}_{n+1}$ fulfils the relation given in (3.12), the quantity $\boldsymbol{\theta}_{n+1}$ is a good estimate of the local error $\boldsymbol{e}_{n+1}$ at time t_{n+1}.

Now, from local estimate $\boldsymbol{\theta}_{n+1}$, we are in condition to build an error estimate for the global error. This important result is introduced in the next theorem.

Theorem 3.2. *Consider a time-stepping method of order p. Then, the sum of error estimates sized in the energy norm and denoted by Θ_n, i.e,*

$$\Theta_n = \sum_{i=1}^{n} ||\boldsymbol{\theta}_i||_E \ . \tag{3.16}$$

is an estimate for the global error $\boldsymbol{E}_n$, that is,

$$||\boldsymbol{E}_n||_E \approx \Theta_n \ . \tag{3.17}$$

PROOF: By Theorem 3.1 and using the triangle inequality and the result provided in (3.15), it follows,

$$
\begin{aligned}
||\boldsymbol{E}_n||_E &\leqslant \sum_{i=1}^{n} ||\boldsymbol{e}_i||_E \\
&= \sum_{i=1}^{n} ||\boldsymbol{\theta}_i + \boldsymbol{\eta}_i||_E \\
&\leqslant \sum_{i=1}^{n} \left[||\boldsymbol{\theta}_i||_E + ||\boldsymbol{\eta}_i||_E\right] \\
&= \sum_{i=1}^{n} \left[||\boldsymbol{\theta}_i||_E + \mathcal{O}(\Delta t^{p+1+k})\right] \\
&\approx \sum_{i=1}^{n} ||\boldsymbol{\theta}_i||_E \\
&= \Theta_n \ .
\end{aligned}
\tag{3.18}
$$

■

Finally, we present a graphical scheme in Figure 3.2 where all quantities defined in this chapter are plotted. In addition, relations which arise between them are also indicated. The aim is to try to clarify the different concepts explained.

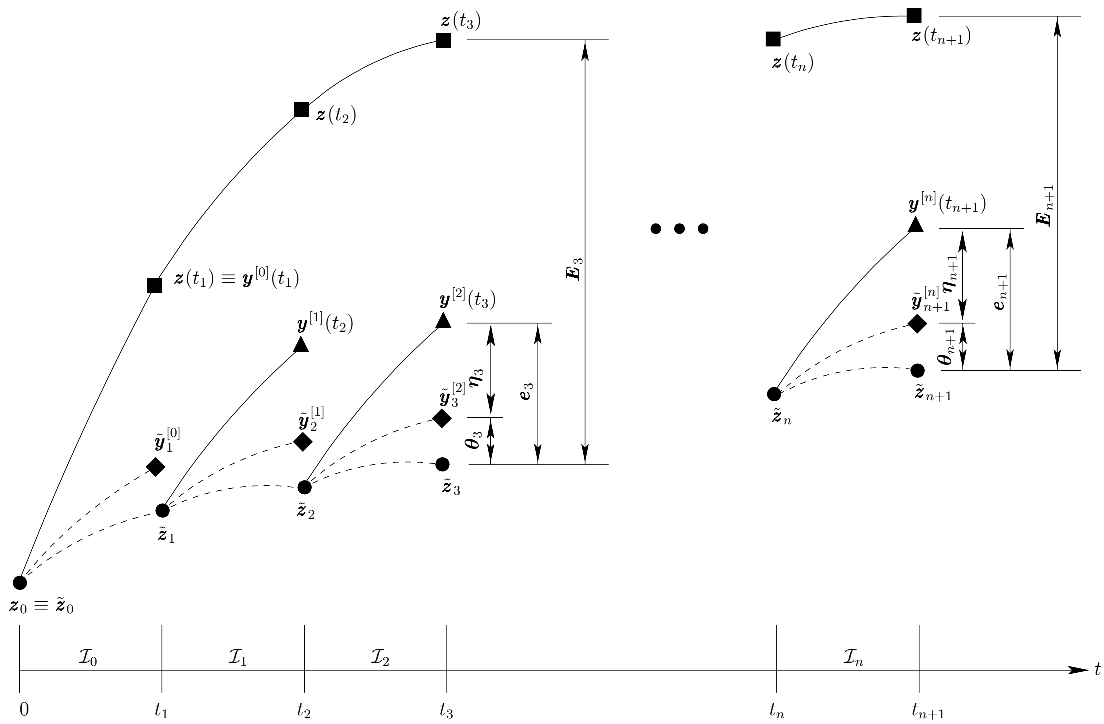


Figure 3.2: Scheme representing the different solutions of a ODE problem and the different errors which have been defined (Continuous lines express exact solutions, whereas dashed lines refer to numerical solutions).

The main conclusion of this section is that the problem of estimating the error is reduced to calculating an *improved solution* to each local problem, whose order of accuracy must be, at least, one more than the numerical solution. For example, for the HHT method this *improved solution* has to be locally fourth-order accurate. Moreover, in the next section it will be seen that the computational cost to obtain it must be small.

3.3 Design of local error estimators: requirements of error estimators

In the previous section, it was shown that a simple way to compute an error estimate is by means of computing an *improved solution*. Now in this section, we address in detail the conditions that this *improved solution* (and, therefore, the error estimator) must satisfy.

These conditions were introduced by ROMERO & LACOMA [2006b] and have a twofold objective. First, they establish a framework where to design new error estimators. Second, they are used for the analysis of estimators existing in the literature in order to investigate their behavior. This analysis will be perform in the next section.

Next, these conditions are presented.

Proposition 3.1. *Consider a p-order accurate time-stepping integration method. Then, the quantity $\boldsymbol{\theta}_{n+1}$ is a useful and valid local error estimator if the following two conditions are satisfied:*

C1. *The quantity $\boldsymbol{\eta}_{n+1}$ associated to $\boldsymbol{\theta}_{n+1}$ via the relation*

$$\boldsymbol{\eta}_{n+1} = \boldsymbol{e}_{n+1} - \boldsymbol{\theta}_{n+1} \ , \tag{3.19}$$

is of size $\mathcal{O}(\Delta t^{p+2})$ or smaller.

C2. *The cost of computing the estimator $\boldsymbol{\theta}_{n+1}$ is small compared to the cost of obtaining a more accurate numerical solution to the local problem.*

Remarks 3.3.

i. If we could compute a solution with a higher order of accuracy than $\tilde{\boldsymbol{z}}_{n+1}$ and at a smaller cost, then immediately, an error estimator could be computed by subtracting this more accurate solution from $\tilde{\boldsymbol{z}}_{n+1}$.

ii. Assume that a p-order method is being used. Then, if we employ an estimator $\boldsymbol{\theta}_{n+1}$ such that the unresolved part $\boldsymbol{\eta}_{n+1}$ is of size $\mathcal{O}(\Delta t^{p+1})$ instead of $\mathcal{O}(\Delta t^{p+2})$, the estimator does not fulfill the bound proved in Theorem 3.2. In other words, we are turning away in our approximation a quantity which is of the same size as the local error itself. An estimate that is based on such approximation can never be accurate. This issue is represented in Figure 3.3. Some estimators of the literature possess this drawback, as we will show in Section 3.4.1. □

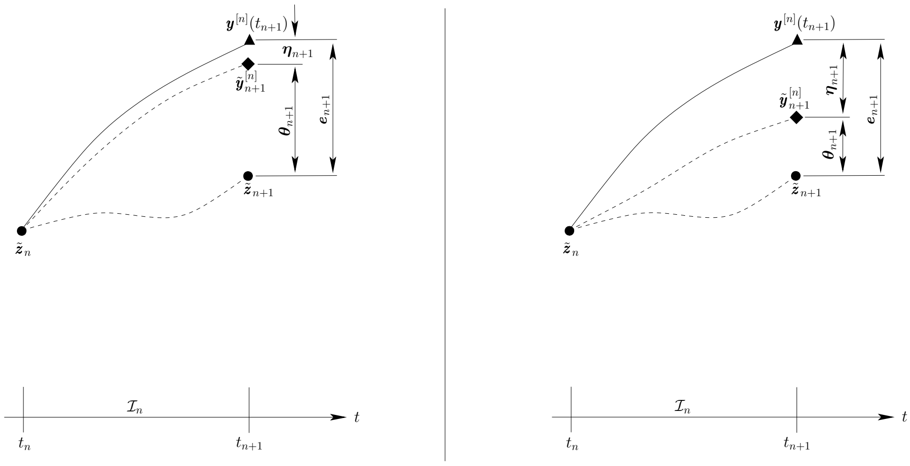


Figure 3.3: Scheme representing two different error estimators. Left: a valid estimator where $\boldsymbol{\theta}_{n+1} \gg \boldsymbol{\eta}_{n+1}$, and therefore $\boldsymbol{e}_{n+1} \approx \boldsymbol{\theta}_{n+1}$. Right: a no valid estimator since $\boldsymbol{\theta}_{n+1} \approx \boldsymbol{\eta}_{n+1}$, and therefore, $\boldsymbol{e}_{n+1} \not\approx \boldsymbol{\theta}_{n+1}$.

In the mathematical literature, different methods for calculating an approximation to $\boldsymbol{y}^{[n]}(t_{n+1})$ have been proposed. One of these methods consists of solving each local problem with two time steps, Δt_n and $\Delta t_n/2$. This procedure is known as Richardson extrapolation and it is described in detail in Hairer et al. [1987], Chapter 2. Another alternative is to employ embedded Runge-Kutta formulas. They use two methods with a different order but that share function evaluations saving a considerable computational cost. The reader is referred to Hairer et al. [1987], Hairer & Wanner [1991], Gear [1971], Press et al. [1992] and Shampine [To appear] for a detailed description of these topics.

On the other hand, in the context of solid and structural dynamics, another procedure has been followed to compute the improved solution $\tilde{\boldsymbol{y}}^{[n]}_{n+1}$. It consists of building the value of the *improved solution* at t_{n+1} by means of an expansion in Taylor's series about t_n.

Following this procedure, in the remainder of section, the condition **C1** is particularized to this case. That is, we investigate what relations each one of the terms of the Taylor's expansion must satisfy.

Before starting the analysis, a useful notation is introduced. Let $\boldsymbol{\vartheta}(t)$ be an arbitrary function of time. We denote by $\boldsymbol{\vartheta}(t_n^-)$ and $\boldsymbol{\vartheta}(t_n^+)$ its limits from the left and from the right at t_n, respectively, i.e.,

$$\boldsymbol{\vartheta}(t_n^-) = \lim_{t \to t_n^-} \boldsymbol{\vartheta}(t) \ , \qquad \boldsymbol{\vartheta}(t_n^+) = \lim_{t \to t_n^+} \boldsymbol{\vartheta}(t) \ . \tag{3.20}$$

This notation will be specially applied to refer to the external load $\mathbf{F}(t)$. Recall that the nth local problem is defined in $[t_n, t_{n+1}]$. Then, variables at t_n are the limits from the right and, at t_{n+1} the limits from the left. However, sometimes, we will use the above notation to emphasize what limit is taken.

To start the analysis, recall that $\boldsymbol{y}^{[n]}(t)$ is the exact solution to the local problem (3.2) in the interval $\mathcal{I}_n$. If $\boldsymbol{y}^{[n]}(t)$ is four times differentiable inside this interval and the external force $\mathbf{F}(t)$ is continuous then, the value of the exact solution at the end of the interval can be expressed as

$$\begin{aligned} \boldsymbol{y}^{[n]}(t_{n+1}) &= \left\{ \begin{array}{l} \boldsymbol{u}^{[n]}(t_{n+1}) \\ \dot{\boldsymbol{u}}^{[n]}(t_{n+1}) \end{array} \right\} \\ &= \left\{ \begin{array}{l} \boldsymbol{u}^{[n]}(t_n) + \Delta t_n \dot{\boldsymbol{u}}^{[n]}(t_n) + \dfrac{\Delta t_n^2}{2} \ddot{\boldsymbol{u}}^{[n]}(t_n^+) + \dfrac{\Delta t_n^3}{6} \dddot{\boldsymbol{u}}^{[n]}(t_n^+) \\ \\ \dot{\boldsymbol{u}}^{[n]}(t_n) + \Delta t_n \ddot{\boldsymbol{u}}^{[n]}(t_n^+) + \dfrac{\Delta t_n^2}{2} \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \dfrac{\Delta t_n^3}{6} \ddddot{\boldsymbol{u}}^{[n]}(t_n^+) \end{array} \right. \\ &\qquad \left. \begin{array}{l} + \dfrac{\Delta t_n^4}{24} \ddddot{\boldsymbol{u}}^{[n]}(t_{n+\xi}) \\ \\ + \dfrac{\Delta t_n^4}{24} \overset{\cdot\cdot\cdot\cdot\cdot}{\boldsymbol{u}}^{[n]}(t_{n+\zeta}) \end{array} \right\} , \end{aligned} \tag{3.21}$$

where $\boldsymbol{u}^{[n]}(t_n)$ are $\dot{\boldsymbol{u}}^{[n]}(t_n)$ are known values, since they are the initial conditions $\mathbf{U}_n$ and $\mathbf{V}_n$, respectively, and the parameters ξ and ζ satisfy $0 \leqslant \xi, \zeta \leqslant 1$. The higher-order rates of local displacements are deduced from the balance equation:

$$\left\{ \begin{array}{l} \ddot{\boldsymbol{u}}^{[n]}(t_n^+) := \mathbf{M}^{-1} \left[\mathbf{F}(t_n^+) - \mathbf{C}\dot{\boldsymbol{u}}^{[n]}(t_n) - \mathbf{K}\boldsymbol{u}^{[n]}(t_n) \right] \ , \\ \dddot{\boldsymbol{u}}^{[n]}(t_n^+) := \mathbf{M}^{-1} \left[\dot{\mathbf{F}}(t_n^+) - \mathbf{C}\ddot{\boldsymbol{u}}^{[n]}(t_n^+) - \mathbf{K}\dot{\boldsymbol{u}}^{[n]}(t_n) \right] \ , \\ \qquad \vdots \\ \text{etc.} \end{array} \right. \tag{3.22}$$

Note that the external force must be continuous within the interval $\mathcal{I}_n$. However, it can have discontinuity points at the start, t_n, or at the end, t_{n+1}, of it. In Section 3.6, this topic will be analyzed with more detail.

In (3.21), if we replace the second and higher order rates by approximations $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$, an *improved solution* at time t_{n+1} is defined:

$$
\begin{aligned}
\tilde{\boldsymbol{y}}_{n+1}^{[n]} &:= \left\{ \begin{array}{c} \tilde{\boldsymbol{u}}_{n+1}^{[n]} \\ \tilde{\boldsymbol{v}}_{n+1}^{[n]} \end{array} \right\} \\
&= \left\{ \begin{array}{c} \mathbf{U}_n + \Delta t_n \mathbf{V}_n + \frac{1}{2}\Delta t_n^2 \tilde{\mathbf{A}}_n + \frac{1}{6}\Delta t_n^3 \tilde{\mathbf{S}}_n \\ \mathbf{V}_n + \Delta t_n \tilde{\mathbf{A}}_n + \frac{1}{2}\Delta t_n^2 \tilde{\mathbf{S}}_n + \frac{1}{6}\Delta t_n^3 \tilde{\mathbf{R}}_n \end{array} \right\} .
\end{aligned}
\tag{3.23}
$$

Now, introducing (3.23) in (3.12), the quantity $\boldsymbol{\eta}_{n+1}$ reads

$$
\begin{aligned}
\boldsymbol{\eta}_{n+1} &= \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}_{n+1}^{[n]} \\
&= \left\{ \begin{array}{c} \dfrac{\Delta t_n^2}{2}\left(\ddot{\boldsymbol{u}}^{[n]}(t_n^+) - \tilde{\mathbf{A}}_n\right) \\ \\ \Delta t_n \left(\ddot{\boldsymbol{u}}^{[n]}(t_n^+) - \tilde{\mathbf{A}}_n\right) \end{array} \right\} + \left\{ \begin{array}{c} \dfrac{\Delta t_n^3}{6}\left(\dddot{\boldsymbol{u}}^{[n]}(t_n^+) - \tilde{\mathbf{S}}_n\right) \\ \\ \dfrac{\Delta t_n^2}{2}\left(\dddot{\boldsymbol{u}}^{[n]}(t_n^+) - \tilde{\mathbf{S}}_n\right) \end{array} \right\} \\
&+ \left\{ \begin{array}{c} \mathcal{O}(\Delta t^4) \\ \\ \dfrac{\Delta t_n^3}{6}\left(\ddddot{\boldsymbol{u}}^{[n]}(t_n^+) - \tilde{\mathbf{R}}_n\right) + \mathcal{O}(\Delta t^4) \end{array} \right\} .
\end{aligned}
\tag{3.24}
$$

Recalling that $\boldsymbol{\eta}_{n+1}$ must be, at least, of size $\mathcal{O}(\Delta t^{p+2})$, the order of accuracy of the approximations to the exact values are easily deduced from the analysis of each term in expression (3.24). These relations are summarized in Table 3.1 for first and second-order accurate methods.

$\boldsymbol{p=1}$	$\boldsymbol{p=2}$
$\tilde{\mathbf{A}}_n = \ddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^2)$	$\tilde{\mathbf{A}}_n = \ddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^3)$
$\tilde{\mathbf{S}}_n = \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t)$	$\tilde{\mathbf{S}}_n = \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^2)$
—	$\tilde{\mathbf{R}}_n = \ddddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t)$

Table 3.1: Conditions to be satisfied by the approximations $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$ depending on the order of accuracy of the numerical method.

Once the *improved solution* given in (3.23) is obtained, the local error estimate at t_{n+1}, $\boldsymbol{\theta}_{n+1}$, is obtained via expression (3.13). For example, for Newmark's method, the local estimate yields:

$$
\begin{aligned}
\boldsymbol{\theta}_{n+1} &= \tilde{\boldsymbol{y}}^{[n]}_{n+1} - \tilde{\boldsymbol{z}}_{n+1} \\
&= \left\{ \begin{array}{c} \dfrac{\Delta t_n^2}{2}\left(\tilde{\mathbf{A}}_n - (1-2\beta)\mathbf{A}_n - 2\beta\mathbf{A}_{n+1}\right) + \dfrac{\Delta t_n^3}{6}\tilde{\mathbf{S}}_n \\ \\ \Delta t_n\left(\tilde{\mathbf{A}}_n - (1-\gamma)\mathbf{A}_n - \gamma\mathbf{A}_{n+1}\right) + \dfrac{\Delta t_n^2}{2}\tilde{\mathbf{S}}_n + \dfrac{\Delta t_n^3}{6}\tilde{\mathbf{R}}_n \end{array} \right\} .
\end{aligned}
\tag{3.25}
$$

Before ending the section, we present some useful results. As discussed in the third item of Remarks 3.1, $\boldsymbol{y}^{[n]}(t_{n+1}) \neq \boldsymbol{y}^{[n+1]}(t_{n+1})$. However, two relations between higher-order rates from different local problems arise. These relations are addressed in the two following lemmas and they will be employed in future proofs.

Lemma 3.1. *Consider a dynamic linear problem as (3.2) integrated numerically in time using a p-order integration method. Then if the external load $\mathbf{F}(t)$ is a continuous function in t_{n+1}, i.e., $\mathbf{F}(t^-_{n+1}) = \mathbf{F}(t^+_{n+1})$, then the following relations holds:*

$$
\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) = \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}) + \mathcal{O}(\Delta t^{p+1}) \ . \tag{3.26}
$$

PROOF: By balance equation $(3.2)_1$ and the consistency of method, the proof reads,

$$
\begin{aligned}
\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) &= \mathbf{M}^{-1}\left[\mathbf{F}(t^-_{n+1}) - \mathbf{C}\dot{\boldsymbol{u}}^{[n]}(t_{n+1}) - \mathbf{K}\boldsymbol{u}^{[n]}(t_{n+1})\right] \\
&= \mathbf{M}^{-1}\left[\mathbf{F}(t^+_{n+1}) - \mathbf{C}\mathbf{V}_{n+1} - \mathbf{K}\mathbf{U}_{n+1}\right] + \mathcal{O}(\Delta t^{p+1}) \\
&= \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}) + \mathcal{O}(\Delta t^{p+1}) \ .
\end{aligned}
\tag{3.27}
$$

■

Lemma 3.2. *Consider a dynamic linear problem as (3.2) integrated numerically in time using the trapezoidal rule. Then if the external load $\mathbf{F}(t)$ is a continuously differentiable function in t_{n+1}, i.e., $\dot{\mathbf{F}}(t^-_{n+1}) = \dot{\mathbf{F}}(t^+_{n+1})$, then the following relation holds:*

$$
\dddot{\boldsymbol{u}}^{[n]}(t_{n+1}) = \dddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \ . \tag{3.28}
$$

PROOF: By equation $(3.22)_2$ and the consistency of the method, it follows,

$$
\begin{aligned}
\dot{\ddot{\boldsymbol{u}}}^{[n]}(t_{n+1}) &= \mathbf{M}^{-1}\left[\dot{\mathbf{F}}(t_{n+1}^{-}) - \mathbf{C}\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) - \mathbf{K}\dot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] \\
&= \mathbf{M}^{-1}\left[\dot{\mathbf{F}}(t_{n+1}^{-}) - \mathbf{C}\mathbf{A}_n - \mathbf{K}\mathbf{V}_{n+1}\right] + \mathcal{O}(\Delta t^3) \\
&= \mathbf{M}^{-1}\left[\dot{\mathbf{F}}(t_{n+1}^{+}) - \mathbf{C}\ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}) - \mathbf{K}\mathbf{V}_{n+1}\right] + \mathcal{O}(\Delta t^3) \\
&= \dot{\ddot{\boldsymbol{u}}}^{[n+1]}(t_{n+1}) + \mathcal{O}(\Delta t^3)\ .
\end{aligned}
\tag{3.29}
$$

■

3.4 Analysis of existing error estimators

3.4.1 Analysis of the local error estimate by ZIENKIEWICZ AND XIE

After introducing the conditions which an error estimator must fulfil, we analyze the error estimator proposed in ZIENKIEWICZ & XIE [1991]. In the original article, this estimator is developed for the Newmark scheme. At the same time, in ZENG ET AL. [1992] an identical expression for the estimator is given, but it is obtained in a different way. In a later work, GÈRADIN & CARDONA [1997] extend it to the HHT method.

An inconvenient of the estimator of ZIENKIEWICZ & XIE [1991] and also of GÈRADIN & CARDONA [1997] is that they only provide an estimate for the error made in displacements, not taking into account the contribution of the velocities. This fact causes that energy norm of the error cannot be taken, and the authors use the L_2-norm. With this norm, the bound defined in (3.6) is not valid, and thus, a bound for the global error cannot be already computed. This will be seen in the numerical simulations in the next chapter.

Following a similar procedure as ZIENKIEWICZ & XIE [1991], HULBERT & JANG [1995] build an estimator for the generalized-α method. In contrast, this one takes into account both the contribution of displacements and velocities. But, when in the generalized-α method a set of parameters are selected such as the Newmark's method arises, the resulting estimator in displacements is the same as in ZIENKIEWICZ & XIE [1991]. Surprisingly, if the selected method is a second-order accurate scheme within the Newmark family, the error in velocities vanishes.

Here, for brevity we restrict our analysis to the error estimator of ZIENKIEWICZ & XIE [1991]. It is defined as follows:

$$\boldsymbol{\theta}^d_{n+1} := \Delta t_n^2 \left(\beta - \frac{1}{6}\right) (\mathbf{A}_{n+1} - \mathbf{A}_n) \ , \tag{3.30}$$

where the superscript d is used to point out that this estimate is only in displacements, and therefore, the vector $\boldsymbol{\theta}^d_{n+1}$ has dimension n_{dof}.

The expression (3.30) is obtained from the following improved solution of the local problem:

$$\tilde{\boldsymbol{y}}^{[n]}_{n+1} := \left\{ \begin{array}{c} \tilde{\boldsymbol{u}}^{[n]}_{n+1} \\ \tilde{\boldsymbol{v}}^{[n]}_{n+1} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{U}_n + \Delta t_n \mathbf{V}_n + \frac{1}{2}\Delta t_n^2 \tilde{\mathbf{A}}_n + \frac{1}{6}\Delta t_n^3 \tilde{\mathbf{S}}_n \\ --- \end{array} \right\} \ , \tag{3.31}$$

where

$$\tilde{\mathbf{A}}_n := \mathbf{A}_n \ , \qquad \tilde{\mathbf{S}}_n := \frac{\mathbf{A}_{n+1} - \mathbf{A}_n}{\Delta t_n} \ , \qquad \tilde{\mathbf{R}}_n = --- \ . \tag{3.32}$$

These relations are completed with the initial value

$$\tilde{\mathbf{A}}_0 := \ddot{\boldsymbol{u}}^{[0]}(0^+) = \mathbf{M}^{-1}\left[\mathbf{F}(0^+) - \mathbf{C}\mathbf{V}_o - \mathbf{K}\mathbf{U}_o\right] = \mathbf{A}_0 \tag{3.33}$$

From (3.30) it is clear that the computational cost of this estimator is very small, satisfying the condition **C2**. Next, we study its accuracy (see condition **C1**). The Newmark family includes first-order methods (when $\gamma \neq 1/2$) and second-order accurate methods (when $\gamma = 1/2$). Depending on the order, the approximations $\tilde{\mathbf{A}}_n$ and $\tilde{\mathbf{S}}_n$ must satisfy different relations (see Table 3.1). We will carry out the analysis in a general case taking a p-order method and then, we will discussed each one of the cases.

First, the order of accuracy of $\tilde{\mathbf{A}}_n$ is analyzed. By working out the value of $\tilde{\mathbf{A}}_n$ in the balance equation $(3.2)_1$, it follows

$$\begin{aligned} \tilde{\mathbf{A}}_n := \mathbf{A}_n &= \mathbf{M}^{-1}\left[\mathbf{F}(t_n) - \mathbf{C}\mathbf{V}_n - \mathbf{K}\mathbf{U}_n\right] \\ &= \mathbf{M}^{-1}\left[\mathbf{F}(t_n) - \mathbf{C}\dot{\boldsymbol{u}}^{[n]}(t_n) - \mathbf{K}\boldsymbol{u}^{[n]}(t_n)\right] \\ &= \ddot{\boldsymbol{u}}^{[n]}(t_n^+) \ . \end{aligned} \tag{3.34}$$

Thus, the vector $\tilde{\mathbf{A}}_n$ always fulfils the conditions established in Table 3.1, whatever it is the order of the method.

Next, the accuracy of $\tilde{\mathbf{S}}_n$ is investigated. Knowing that the algorithmic acceleration $\mathbf{A}_{n+1}$ is an approximation of order $\mathcal{O}(\Delta t^{p+1})$ to $\ddot{\boldsymbol{u}}^{[n]}(t_{n+1})$ (HULBERT

& Hughes [1987]) and using the equality (3.34), it follows

$$
\begin{aligned}
\tilde{\mathbf{S}}_n := \frac{\mathbf{A}_{n+1} - \mathbf{A}_n}{\Delta t_n} &= \frac{\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_n^+)}{\Delta t_n} \\
&= \frac{\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) - \ddot{\boldsymbol{u}}^{[n]}(t_n^+)}{\Delta t_n} + \mathcal{O}(\Delta t^p) \\
&= \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta t^p) \\
&= \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t) \ .
\end{aligned}
\tag{3.35}
$$

From (3.35) it is concluded that in the case of having a first-order method, the vector $\tilde{\mathbf{S}}_n$ is sufficiently accurate, since it is indeed an approximation of order $\mathcal{O}(\Delta t)$ to $\dddot{\boldsymbol{u}}^{[n]}(t_n^+)$. However, for a second-order method, it does not hold. In this case, the order of approximation of $\tilde{\mathbf{S}}_n$ keeps being $\mathcal{O}(\Delta t)$ when it should be one order more accurate, i.e., $\mathcal{O}(\Delta t^2)$ (see Table 3.1) . This fact will discussed again in the numerical simulations in next chapter.

After the analysis of error estimator proposed by Zienkiewicz & Xie [1991], we conclude that this estimator is only valid for estimating the error in displacements when a first-order method is used, since otherwise it will not provide safe error bounds.

3.4.2 Analysis of the local error estimate by Wiberg and Li

Now, we study the error estimator proposed in Wiberg & Li [1993]. As will be seen later, this estimator takes into account the error made in both displacements and velocities. Although in the original paper the authors build it for the Newmark's method, here in order to make the development clearer we restrict our analysis to the particular case of the trapezoidal rule ($\gamma = 1/2$ and $\beta = 1/4$). This is one of the most famous implicit schemes of the Newmark family and it is second-order accurate.

In the remainder of section, we prove that the estimator of Wiberg & Li [1993] works correctly whenever the solution of the problem is very smooth. However, in those problems where the solution is not very smooth, this estimator provides a bound excessively large.This drawback is related to the degree of differentiability of the function which defines the external forces.

Using the notation presented in Section 3.3, the estimator of Wiberg & Li

[1993] is obtained from the following *improved solution*:

$$\tilde{\boldsymbol{y}}^{[n]}_{n+1} := \left\{ \begin{array}{c} \tilde{\boldsymbol{u}}^{[n]}_{n+1} \\ \tilde{\boldsymbol{v}}^{[n]}_{n+1} \end{array} \right\} = \left\{ \begin{array}{l} \mathbf{U}_n + \Delta t_n \mathbf{V}_n + \frac{1}{2}\Delta t_n^2 \tilde{\mathbf{A}}_n + \frac{1}{6}\Delta t_n^3 \tilde{\mathbf{S}}_n + \frac{1}{24}\Delta t_n^4 \tilde{\mathbf{R}}_n \\ \mathbf{V}_n + \Delta t_n \tilde{\mathbf{A}}_n + \frac{1}{2}\Delta t_n^2 \tilde{\mathbf{S}}_n + \frac{1}{6}\Delta t_n^3 \tilde{\mathbf{R}}_n \end{array} \right\} , \tag{3.36}$$

where the approximations $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$ are defined by:

$$\tilde{\mathbf{A}}_n := \mathbf{A}_n \ , \quad \tilde{\mathbf{S}}_n := 2\frac{\mathbf{A}_n - \mathbf{A}_{n-1}}{\Delta t_{n-1}} - \tilde{\mathbf{S}}_{n-1} \ , \quad \tilde{\mathbf{R}}_n := \frac{\tilde{\mathbf{S}}_{n+1} - \tilde{\mathbf{S}}_n}{\Delta t_n} \ . \tag{3.37}$$

In order to define completely these formulas, the initial values are the following:

$$\tilde{\mathbf{A}}_0 := \ddot{\boldsymbol{u}}^{[0]}(0^+) \ , \qquad \tilde{\mathbf{S}}_0 := \dddot{\boldsymbol{u}}^{[0]}(0^+) \ , \tag{3.38}$$

recalling that the initial rates are computed by derivating the balance equation, i.e.,

$$\begin{aligned} \ddot{\boldsymbol{u}}^{[0]}(0^+) &:= \mathbf{M}^{-1}\left[\mathbf{F}(0^+) - \mathbf{C}\mathbf{V}_o - \mathbf{K}\mathbf{U}_o\right] \ , \\ \dddot{\boldsymbol{u}}^{[0]}(0^+) &:= \mathbf{M}^{-1}\left[\dot{\mathbf{F}}(0^+) - \mathbf{C}\ddot{\boldsymbol{u}}^{[0]}(0^+) - \mathbf{K}\mathbf{V}_o\right] \ . \end{aligned} \tag{3.39}$$

The analysis follows the same procedure than before. From (3.36) and (3.37) it follows that this estimator is cheap to compute and, therefore, the condition **C2** holds.

Next, we study of order of accuracy of the *improved solution* defined in (3.36). For this, first, the vector $\tilde{\mathbf{A}}_n$ is investigated. As can be seen, this approximation is equal to that employed by ZIENKIEWICZ & XIE [1991] in the previous section and whose proof was given in (3.34).

The second approximation $\tilde{\mathbf{S}}_n$ is proved by induction. By definition, the first term of the series is $\tilde{\mathbf{S}}_0 := \dddot{\boldsymbol{u}}^{[0]}(0^+)$ and obviously the condition established in Table 3.1 holds. Now, one assumes this condition is fulfilled at time t_n, and studies the following term. Then, using the order of consistency of the trapezoidal rule, relation given in (3.34) and Taylor's expansions about t_n, it follows that

$$\begin{aligned} \tilde{\mathbf{S}}_{n+1} &:= 2\frac{\mathbf{A}_{n+1} - \mathbf{A}_n}{\Delta t_n} - \tilde{\mathbf{S}}_n = 2\frac{\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) - \ddot{\boldsymbol{u}}^{[n]}(t_n^+)}{\Delta t_n} - \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^2) \\ &= 2\frac{\ddot{\boldsymbol{u}}^{[n]}(t_n^+) + \Delta t_n \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \frac{\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[n]}(t_n^+) - \ddot{\boldsymbol{u}}^{[n]}(t_n^+)}{\Delta t_n} - \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^2) \\ &= \dddot{\boldsymbol{u}}^{[n]}(t_n^+) + \Delta t_n \ddddot{\boldsymbol{u}}^{[n]}(t_n^+) + \mathcal{O}(\Delta t^2) \\ &= \dddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^2) \ . \end{aligned} \tag{3.40}$$

Development performed in (3.40) does not suffice to prove the order of accuracy of $\tilde{\mathbf{S}}_{n+1}$, since it demands to known the size of the difference $\tilde{\mathbf{S}}_{n+1} - \dot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+})$. However, using Lemma 3.2, it is concluded that

$$\tilde{\mathbf{S}}_{n+1} = \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^2) = \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+}) + \mathcal{O}(\Delta t^2) \ . \tag{3.41}$$

Recall that Lemma 3.2 requires that the external load $\mathbf{F}(t)$ is *continuously differentiable*. That is, the value $\mathbf{S}_n$ has the desired accuracy if and only if the external force is a *continuously differentiable* function.

Finally, it remains to prove the order of the fourth order rate $\tilde{\mathbf{R}}_n$. We proceed again by induction. This proof is very similar to that followed in (3.40) and (3.41). First, it is studied the initial term $\tilde{\mathbf{R}}_0$ and then, assuming that the condition holds for $\tilde{\mathbf{R}}_n$, the term $\tilde{\mathbf{R}}_{n+1}$ is examined. Thus, using the order of approximation of $\tilde{\mathbf{S}}_n$, the results addressed in Lemma 3.2 and Taylor's expansions about t_n, it follows that:

- For $n = 0$:

$$\begin{aligned}
\tilde{\mathbf{R}}_0 := \frac{\tilde{\mathbf{S}}_1 - \tilde{\mathbf{S}}_0}{\Delta t_0} &= \frac{\ddot{\boldsymbol{u}}^{[1]}(t_1^{+}) - \ddot{\boldsymbol{u}}^{[0]}(0^{+})}{\Delta t_0} + \mathcal{O}(\Delta t) \\
&= \frac{\ddot{\boldsymbol{u}}^{[0]}(t_1^{+}) - \ddot{\boldsymbol{u}}^{[0]}(0^{+})}{\Delta t_0} + \mathcal{O}(\Delta t) \\
&= \frac{\ddot{\boldsymbol{u}}^{[0]}(0^{+}) + \Delta t_0 \dddot{\boldsymbol{u}}^{[0]}(0^{+}) - \ddot{\boldsymbol{u}}^{[0]}(0^{+})}{\Delta t_0} + \mathcal{O}(\Delta t) \\
&= \dddot{\boldsymbol{u}}^{[0]}(0^{+}) + \mathcal{O}(\Delta t) \ .
\end{aligned} \tag{3.42}$$

- For $n+1$:

$$\begin{aligned}
\tilde{\mathbf{R}}_{n+1} := \frac{\tilde{\mathbf{S}}_{n+2} - \tilde{\mathbf{S}}_{n+1}}{\Delta t_{n+1}} &= \frac{\ddot{\boldsymbol{u}}^{[n+2]}(t_{n+2}^{+}) - \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+})}{\Delta t_{n+1}} + \mathcal{O}(\Delta t) \\
&= \frac{\ddot{\boldsymbol{u}}^{[n+1]}(t_{n+2}^{+}) - \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+})}{\Delta t_{n+1}} + \mathcal{O}(\Delta t) \\
&= \frac{\ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+}) + \Delta t_{n+1} \dddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+}) - \ddot{\boldsymbol{u}}^{[n+1]}(t_{n+1}^{+})}{\Delta t_{n+1}} + \mathcal{O}(\Delta t) \\
&= \dddot{\boldsymbol{u}}^{[n+1]}(\Delta t_{n+1}^{+}) + \mathcal{O}(\Delta t) \ .
\end{aligned} \tag{3.43}$$

Therefore, this estimator only provides accurate error estimates when the external load is continuously differentiable. The same conclusion is reached for first-order methods.

We must note that, from the theory of ODEs. the initial value problem (2.1) has a unique solution if the external force $\mathbf{F}(t)$ is continuous. The requirement of continuity of derivates is unnecessary for existence and uniqueness of the solution and, more importantly, rules out important applications which are perfectly defined but fail to have differentiable loads.

3.5 A new simple error estimator

In the previous section we have analyzed two existing and widely employed error estimators for structural and solid dynamic problems. We have proved that the estimator of ZIENKIEWICZ & XIE [1991] neglects the contribution of error in velocities to the total error. In addition, the quantity $\boldsymbol{\eta}_{n+1}$ fails to satisfy the needed conditions when second-order methods are employed. Also, the estimator of WIBERG & LI [1993] has a restricted range of applicability.

Several real problems involve non-smooth loads. For instance, problems with impulsive loads, triangular loads, sawtooth loads, etc. Also, in non-linear problems these non-smooth forces may appear in those situations where contacts between different bodies or between parts of the same body take place.

To solve the aforementioned limitations, a new error estimator is developed in this section. For brevity, we focus on the particular case of second-order accurate methods of the Newmark family. This estimator stands out by its simplicity and it is easily implemented in existing finite element programs.

The objective can be summarized as finding accurate approximations $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$ at a small computational cost. The most simple way to compute higher order rates is by differentiating the balance equation $(3.2)_1$, as mentioned in (3.22). As can be seen, in order to compute the third-order rate and the successive ones is necessary to know the rates of the external forces. However, recalling the conditions summarized in Table 3.1, for a second-order accurate integration method, it suffices to calculate second and first-order approximations to the third and fourth-order rates, respectively. Therefore, denoting by $\tilde{\mathbf{F}}_n^1$ a second-order approximation to $\dot{\mathbf{F}}(t_n^+)$, and by $\tilde{\mathbf{F}}_n^2$ a first-order one to $\ddot{\mathbf{F}}(t_n^+)$, i.e.,

$$\begin{aligned} \tilde{\mathbf{F}}_n^1 &= \dot{\mathbf{F}}(t_n^+) + \mathcal{O}(\Delta t^2) \ , \\ \tilde{\mathbf{F}}_n^2 &= \ddot{\mathbf{F}}(t_n^+) + \mathcal{O}(\Delta t) \ , \end{aligned} \tag{3.44}$$

then, values for $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$ are given by:

$$\begin{cases} \tilde{\mathbf{A}}_n := \mathbf{A}_n \ , \\ \tilde{\mathbf{S}}_n := \mathbf{M}^{-1} \left[\tilde{\mathbf{F}}^1_n - \mathbf{C}\mathbf{A}_n - \mathbf{K}\mathbf{V}_n \right] \ , \\ \tilde{\mathbf{R}}_n := \mathbf{M}^{-1} \left[\tilde{\mathbf{F}}^2_n - \mathbf{C}\tilde{\mathbf{S}}_n - \mathbf{K}\tilde{\mathbf{A}}_n \right] \ . \end{cases} \tag{3.45}$$

It is immediate to check that approximations (3.45) together with (3.44) satisfy the relation given in Table 3.1 and, therefore, this estimator fulfils the condition **C1**.

The two approximations indicated in (3.44) can be computed by means of differentiating numerically $\mathbf{F}(t)$ at some $t = t_n + \epsilon\Delta t_n$, with $\epsilon \ll \Delta t_n$. Two valid formulas for $\tilde{\mathbf{F}}^1_n$ and $\tilde{\mathbf{F}}^1_n$ are:

$$\begin{aligned} \tilde{\mathbf{F}}^1_n &= \frac{4\mathbf{F}(t_n+\epsilon) - \mathbf{F}(t_n+2\epsilon) - 3\mathbf{F}(t_n)}{2\epsilon} \ , \\ \tilde{\mathbf{F}}^2_n &= \frac{\mathbf{F}(t_n+2\epsilon) - \mathbf{F}(t_n+\epsilon) + \mathbf{F}(t_n)}{\epsilon^2} \ , \end{aligned} \tag{3.46}$$

where a suitable value for ϵ is $0.001 \cdot \Delta t_n$.

Remarks 3.4.

i. The computational cost of this new estimator is determined by the cost of solving the linear system of equations $(3.45)_2$ and $(3.45)_3$, where the matrix is always the mass matrix. Then, by factorizing this matrix (e.g, a LU factorization) at the beginning of the analysis, the cost of the estimator remains small.

ii. The estimator presented in this section has been developed for second-order methods within the Newmark family (when $\gamma = 1/2$). For other time-stepping methods, such us the HHT or the generalized-α, the followed scheme is not directly applicable, since numerical accelerations are not second-order accurate. Therefore, in order to preserve the accuracy of the estimator, a new acceleration must be recomputed in each time step. This fact will be widely discussed in next chapter (Section 4.4.1), where different alternatives to overcome this drawback will be shown.

iii. In the case of first-order methods ($\gamma \neq 1/2$), the approximation $\tilde{\mathbf{R}}_n$ vanishes and the order of accuracy of $\tilde{\mathbf{F}}^1_n$ can be decrease up to order $\mathcal{O}(\Delta t)$. □

3.5.1 Summary of the implementation of error estimator

The algorithm to compute the new error estimator for the Newmark's method is summarized in Box 3.1.

Given the solution at t_n, that is, the vectors $\mathbf{U}_n$, $\mathbf{V}_n$, $\mathbf{A}_n$:

1. Compute the solution at t_{n+1}, $\mathbf{U}_{n+1}$, $\mathbf{V}_{n+1}$, $\mathbf{A}_{n+1}$ using the Newmark's method.

2. Compute the auxiliary vectors $\tilde{\mathbf{S}}_n$ and $\tilde{\mathbf{R}}_n$:

$$\tilde{\mathbf{S}}_n := \mathbf{M}^{-1}\left[\tilde{\mathbf{F}}_n^1 - \mathbf{C}\mathbf{A}_n - \mathbf{K}\mathbf{V}_n\right] \ , \tag{3.47}$$

$$\tilde{\mathbf{R}}_n := \mathbf{M}^{-1}\left[\tilde{\mathbf{F}}_n^2 - \mathbf{C}\tilde{\mathbf{S}}_n - \mathbf{K}\mathbf{A}_n\right] \ . \tag{3.48}$$

3. Compute the error estimate:

$$\boldsymbol{\theta}_{n+1} = \left\{ \begin{array}{c} \beta\Delta t_n^2(\mathbf{A}_n - \mathbf{A}_{n+1}) + \frac{\Delta t_n^3}{6}\tilde{\mathbf{S}}_n \\ \gamma\Delta t_n(\mathbf{A}_n - \mathbf{A}_{n+1}) + \frac{\Delta t_n^2}{2}\tilde{\mathbf{S}}_n + \frac{\Delta t_n^3}{6}\tilde{\mathbf{R}}_n \end{array} \right\} \ . \tag{3.49}$$

4. Compute its energy norm:

$$\Theta_{n+1} := ||\boldsymbol{\theta}_{n+1}||_E \ . \tag{3.50}$$

5. Set $n \leftarrow n+1$:

$$\begin{array}{l} \texttt{IF } t_n < T \texttt{ THEN GO TO } \mathbf{1}. \\ \texttt{IF } t_n = T \texttt{ EXIT} \end{array} \tag{3.51}$$

Box 3.1: Summary of the implementation of the local error estimate for Newmark's method

3.6 Discontinuities in the external load

The analysis presented in Sections 3.3-3.5 requires that external force $\mathbf{F}(t)$ is continuous in the interior of the interval $\mathcal{I}_n$. Next, we study those cases where $\mathbf{F}(t)$ has points of discontinuity.

Consider an external load such as the one represented in Figure 3.4. which is discontinuous within of interval of interest. An IVP (2.1) excited with this type of load has not a unique solution, since the continuity of the external force is a necessary condition to ensure the uniqueness of solution (see any classical text of differential equations).

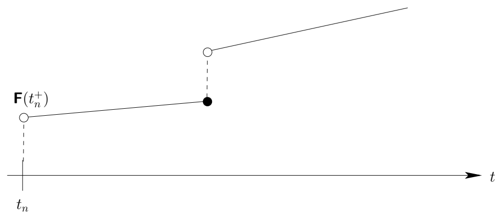


Figure 3.4: A discontinuous external load

If we want to solve problems like that capturing the discontinuity, we can split the problem in two and to do coincide the discrete instants t_i with points where the external load is discontinuous, as depicted in Figure 3.5.

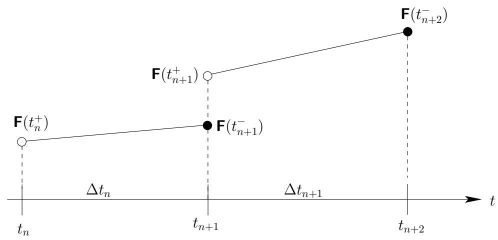


Figure 3.5: Time steps to capture a discontinuity.

Then, the analysis is performed until the point of discontinuity. At this point, the analysis is stopped and the acceleration must be recomputed taking into account $\mathbf{F}(t_{n+1}^+)$, i.e.,

$$\mathbf{A}_{n+1} = \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1}^+) - \mathbf{C}\mathbf{V}_{n+1} - \mathbf{K}\mathbf{U}_{n+1}\right] \ . \tag{3.52}$$

With this acceleration, we can start again the analysis until the next discontinuity point. Following this procedure, the discontinuities will be located in the extremes of the time intervals and, therefore, the solution will be continuous in each local problem. Then, the analysis presented in this chapter is perfectly applicable in this type of problems.

3.7 Representative numerical simulations

In this section we present several numerical simulations which will allow us to discuss some of conclusions addressed in Sections 3.4 and 3.5.

The problem analyzed is a simple one, but where important facts arise. It is a spring-mass system with one degree of freedom and with the following data: mass $\mathbf{M} = 1$ kg, stiffness $\mathbf{K} = 6$ N/m and initial displacement $\mathbf{U}(0) = 1$ m. With these data the equation of motion of this system is:

$$\begin{cases} \ddot{\mathbf{U}}(t) + 6\mathbf{U}(t) = \mathbf{F}(t), \qquad t \in [0,5], \\ \mathbf{U}(0) = 1 \ , \\ \dot{\mathbf{U}}(0) = 0 \ . \end{cases} \tag{3.53}$$

Two different situations are considered. In the first one, the external load which excites the system is smooth, specifically, a sinusoidal load. In the second case, a triangular load is applied to the system resulting in a non-smooth response. All simulations are carried out using the trapezoidal rule and the equation is integrated in time up to 5 s. In each case we compare the results provided by the new error estimator introduced in Section 3.5 to those obtained by the error estimator of WIBERG & LI [1993].

Also, the estimated global errors are compared to the exact ones, since the exact solution of the problem can be easily computed.

3.7.1 Sinusoidal load

In this first case, the external load applied to the system is a sinusoidal load of frequency 2π rad/s, i.e.,

$$\mathbf{F}(t) = \sin(2\pi t), \qquad t \in [0,5] \ . \tag{3.54}$$

Figure 3.6 depicts the exact local error $\boldsymbol{e}_{n+1}$ and the quantity $\boldsymbol{\eta}_{n+1}$ for the estimator of WIBERG & LI [1993]. More specifically, the two first graphs shown these quantities but split in their contribution in displacements and velocities and sized in the L_2-norm. The third graph represents the complete quantities in the energy norm. We observe that the *improved solution* $\tilde{\boldsymbol{y}}_{n+1}^{[n]}$ is fifth-order accurate in displacements and fourth-order accurate in both velocities and energy norm, and therefore, condition **C1** established in Proposition 3.1 is satisfied.

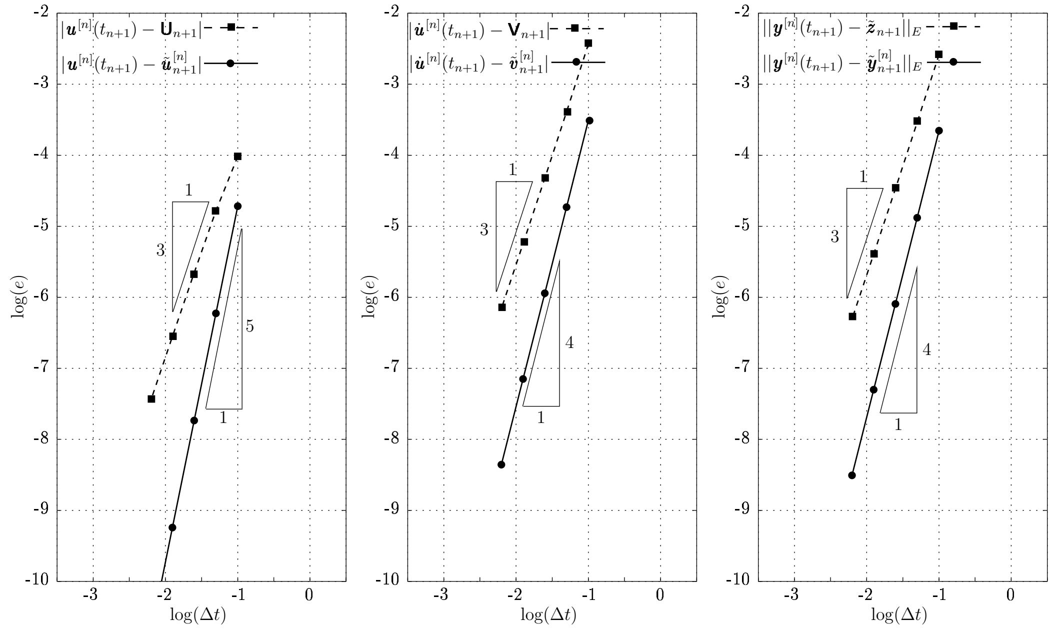


Figure 3.6: Rate of convergence of estimated local errors at time $t = 5$ s. Results obtained with the error estimator of WIBERG & LI [1993] (Left: displacement error; Center: velocity error; Right: total error).

In Figure 3.7, the same quantities are plotted but now for the new error estimator proposed in Section 3.5. Again, the *improved solution* has the desired accuracy.

Figure 3.8 shows the evolution in time of the local errors for both estimators, when a constant time step $\Delta t = 0.05$ s is used. The results are normalized by dividing by the square root of the initial mechanical energy of the system. Both estimates have a very similar behavior.

Adding up these local errors as explained in Theorem 3.2 an estimate for the global error is computed. The evolution of this global estimate is plotted in Figure 3.9 for both estimators, and these are compared to the exact error. We observe that both estimators provide accurate upper bounds.

Finally, the rate of convergence of global estimated errors is investigated. It is desirable that they exhibit the same rate than the exact error, in this case (trapezoidal rule) a quadratic rate. Figure 3.10 confirms this behavior.

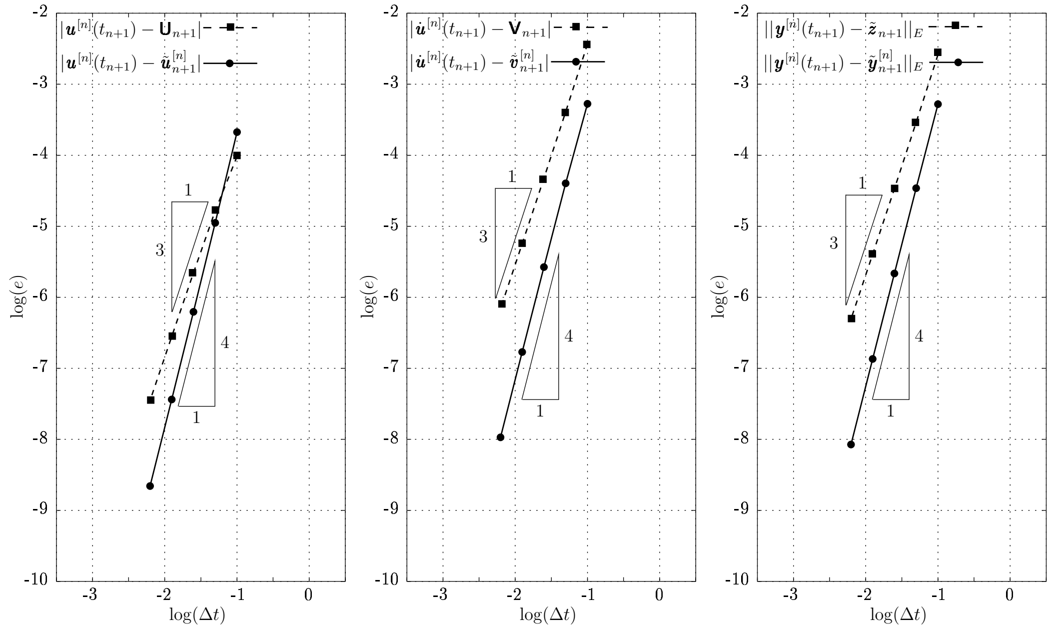


Figure 3.7: Rate of convergence of estimated local errors at time $t = 5$ s. Results obtained with the new error estimator (Left: displacement error; Center: velocity error; Right: total error).

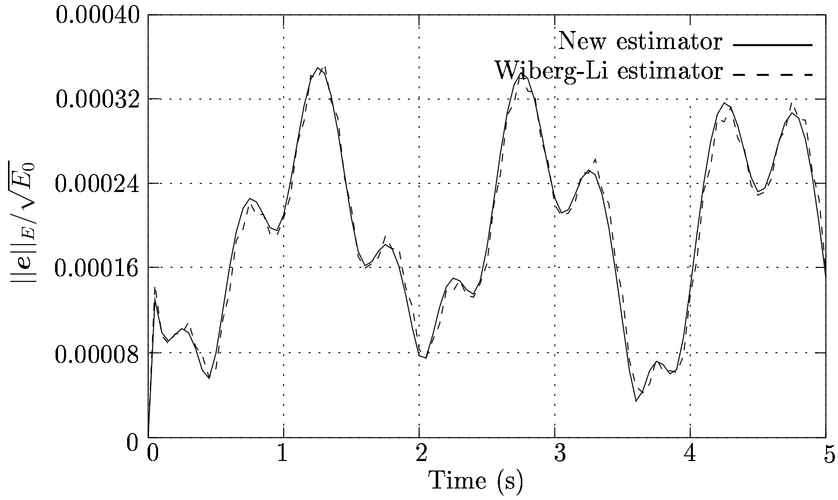


Figure 3.8: Local error estimates in energy norm with time step $\Delta t = 0.05$ s for sinusoidal load. E_0 is the initial energy of the system.

The conclusion of this first simulation is that when the dynamic response is smooth, both the estimator of Wiberg & Li [1993] and the new one provide very similar estimates which are accurate error bounds. However, in the next example where the response in not smooth, important differences arise.

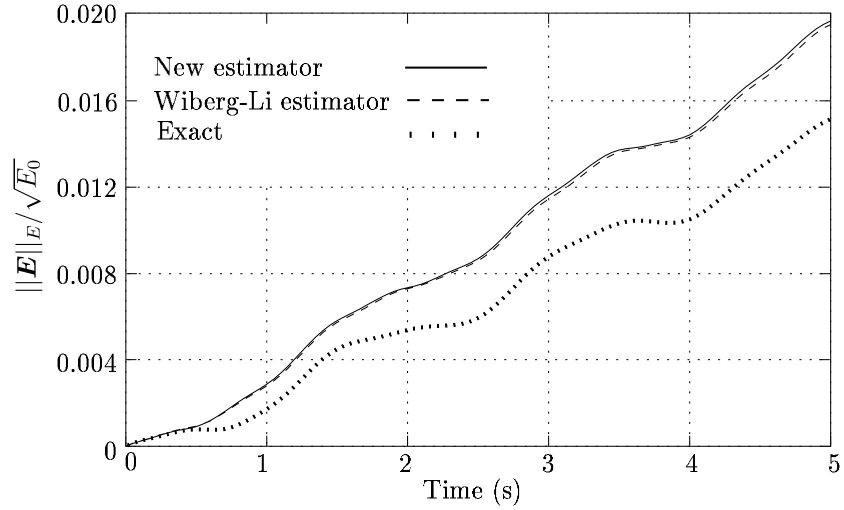


Figure 3.9: Global error estimates in energy norm with time step $\Delta t = 0.05$ s for sinusoidal load. E_0 is the initial energy of the system.

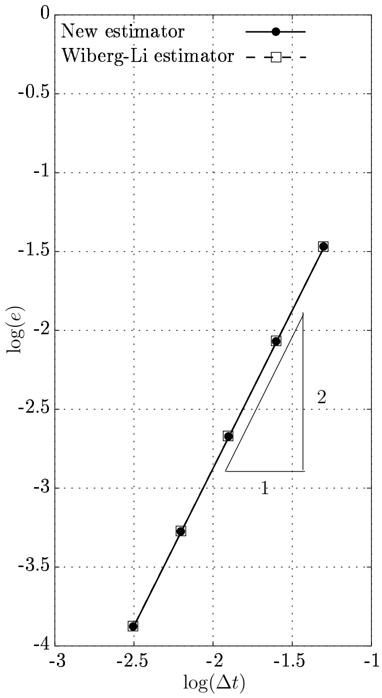


Figure 3.10: Rate of convergence of the estimated global errors at time $t = 5.0$ s for sinusoidal load.

3.7.2 Triangular load

In this second case, the mass-spring system is excited with a non-smooth external load: a triangular-shaped load with an amplitude of 1 N and frequency of 1 Hz. Note that this loading function is continuous everywhere but at time instants

$$t = 0.5i - 0.25, \qquad i = 1, 2, 3, \dots\,, \tag{3.55}$$

it is not differentiable.

Figure 3.11 depicts the evolution of estimated local errors. It can be seen that at instants where the loading function is not differentiable, the estimator of Wiberg & Li [1993] exhibits sharp changes in the value of its error estimation. However, the new proposed estimator provides a smoother response. The abrupt changes in the estimation of the local error by the estimator of Wiberg & Li [1993] causes an important overestimation of the global error (see Figure 3.12). On the contrary, the new one computed an estimation very close to the exact error. This fact agrees with the conclusion reached in Section 3.4.2 (recall that we concluded error estimator of Wiberg & Li [1993] fails to satisfy the necessary conditions when the external load is not continuously differentiable).

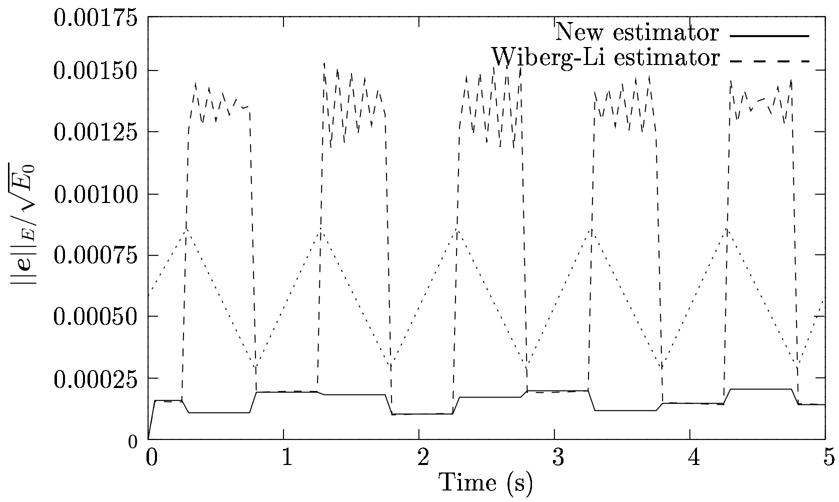


Figure 3.11: Relative local error estimates in energy norm with time step $\Delta t = 0.05$ s for triangular load. E_0 is the initial energy of the system. A plot of the loading function is superimposed with dots so that the instants at which the loading is not differentiable can be easily identified.

When the order a accuracy of the estimators is investigated, we observe that estimator of Wiberg & Li [1993] loses one order, being now of $\mathcal{O}(\Delta t)$. In

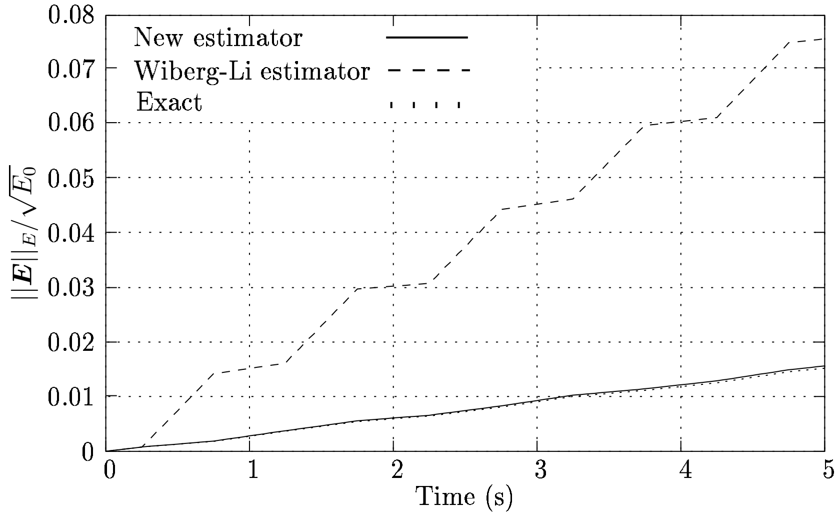


Figure 3.12: Global error estimates in energy norm with time step $\Delta t = 0.05$ s for triangular load. E_0 is the initial energy of the system.

contrast, as in the smooth loading case, the new estimator remains the correct order. It can be checked in Figure 3.12.

In Section 3.4.2 we proved that, in problems like this, the third-order approximation $\tilde{\mathbf{S}}_n$ which uses the estimator of Wiberg & Li [1993] have shortcomings. In Figure 3.14, this third-order rate is depicted and it is compared to the one computed by the new estimator, plotted in Figure 3.15. It is clear that the former estimator provides a third-order rate with a very oscillatory behavior in contrast to the one computed by the later.

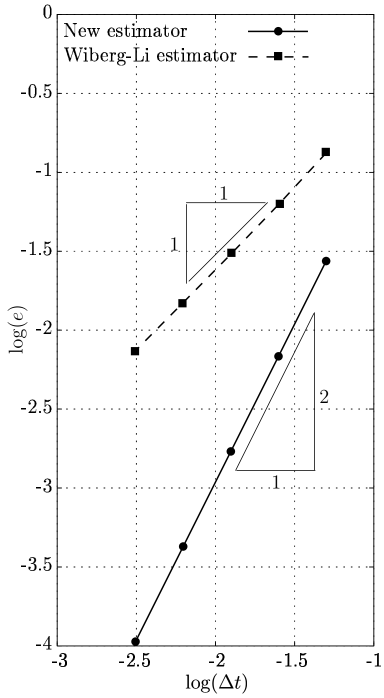


Figure 3.13: Rate of convergence of the estimated global errors at time $t = 5.0$ s for triangular load.

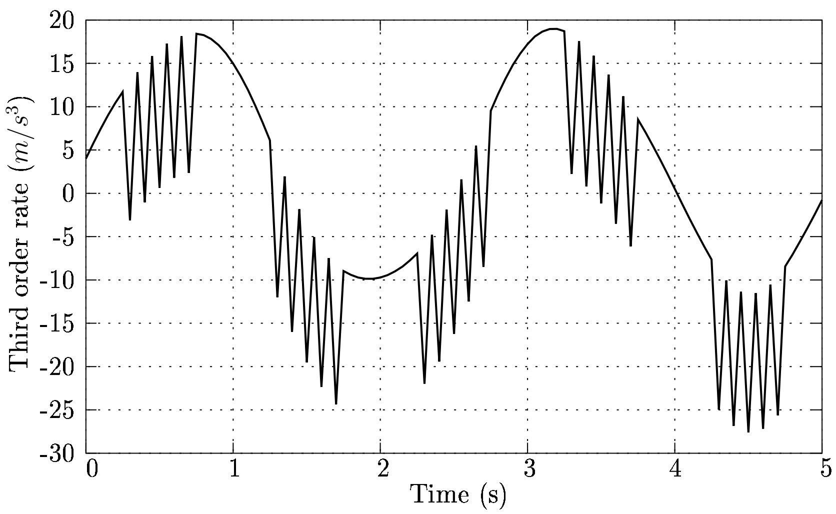


Figure 3.14: Evolution of the third order rate $\tilde{\mathbf{S}}_n$ computed by the error estimator of Wiberg & Li [1993]

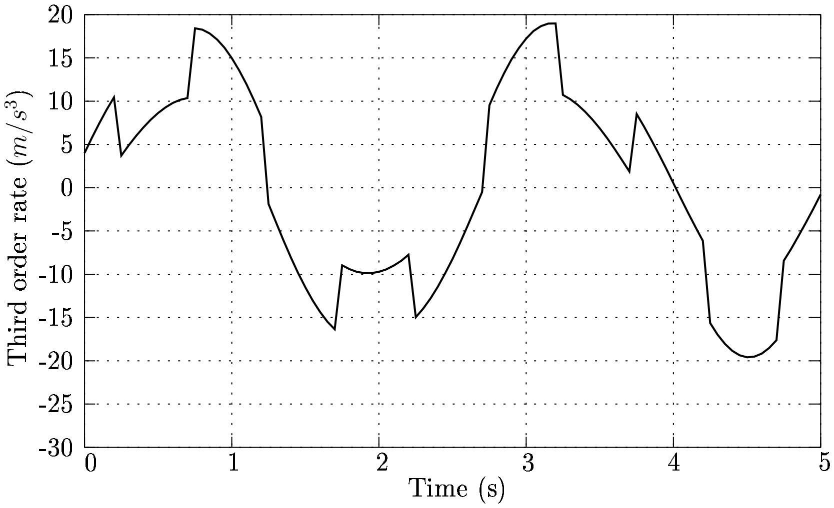


Figure 3.15: Evolution of the third order rate $\tilde{\mathbf{S}}_n$ computed by the new error estimator.

Chapter 4

A new strategy for designing error estimators in the context of solid dynamics

After analyzing in the previous chapter some existing error estimators, the goal of this chapter is to develop a new general estimator to be used with Newmark's and the HHT method. This new estimator is built following a novel procedure and provide accurate estimates in all type of situations.

An outline of the chapter is as follows. Section 4.1 reviews some concepts on error estimation which were discussed in Chapter 3. In Section 4.2 we present a new strategy for the design of error estimators for time-stepping methods. In Section 4.3, this strategy is particularized to the Newmark family and in Section 4.4 to the HHT method. As will be seen, in the case of the HHT method, it is mandatory to compute a new acceleration since the one computed by this method has not the required accuracy. This issue is largely discussed in Section 4.4. Finally, Section 4.5 illustrates the performance of this new estimator in several numerical simulations.

4.1 A review on error estimation in dynamic problems

As discussed in the previous chapter, and to the author's knowledge, all error estimators specifically designed in the context of solid and structural dynamics are obtained on the basis of Taylor series' expansions of the solution. Therefore, these estimators need to approximate displacement rates higher than second-order. A first alternative to compute these rates consists of approximating them using known values of accelerations. This is the procedure followed by the estimator of ZIENKIEWICZ & XIE [1991] and the ones derived from it (CHOI & CHUNG [1996]; CHUNG ET AL. [2003]; GÈRADIN & CARDONA [1997]; HULBERT & JANG [1995]; ZENG ET AL. [1992]). In Section 3.4.1, we proved that this class of estimators does not work correctly with second-order time-stepping methods.

A second alternative is to compute some of this rates exactly by derivating the balance equation and the remaining ones using approximations as explained above. One example is the estimator of WIBERG & LI [1993]. Recall that it computes exactly the initial third-order rate $\tilde{\mathbf{S}}_0$. With this vector and the accelerations, the successive third and fourth-order rates are approximated. It was shown that this method demands a certain smoothness in the external loading function to provide accurate estimates.

The last alternative is to compute all higher rates exactly, as the new estimator proposed in Section 3.5. This strategy is trivial in linear problems. However, if we want to extent it to non-linear simulations, then, the method is not so simple. For example, in a non-linear case the third-order local rate reads:

$$\dddot{\boldsymbol{u}}^{[n]}(t_n) := \mathbf{M}^{-1}\left[\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{F}\left(t_n, \boldsymbol{u}^{[n]}(t_n)\right) - \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{F}_{\text{int}}\left(\boldsymbol{u}^{[n]}(t_n), \dot{\boldsymbol{u}}^{[n]}(t_n)\right)\right] \ , \tag{4.1}$$

where the temporal derivative of the external forces is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{F}\left(t_n, \boldsymbol{u}^{[n]}(t_n)\right) &= \frac{\partial}{\partial t}\mathbf{F}\left(t_n, \boldsymbol{u}^{[n]}(t_n)\right) + \frac{\partial \mathbf{F}}{\partial \boldsymbol{u}^{[n]}}\dot{\boldsymbol{u}}^{[n]}(t_n) \\ &= \frac{\partial}{\partial t}\mathbf{F}\left(t_n, \boldsymbol{u}^{[n]}(t_n)\right) + \mathbf{K}_{\text{ext}}\dot{\boldsymbol{u}}^{[n]}(t_n) \ ,\end{aligned} \tag{4.2}$$

and the temporal derivative of the internal forces is given by

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{F}_{\text{int}}\left(\boldsymbol{u}^{[n]}(t_n), \dot{\boldsymbol{u}}^{[n]}(t_n)\right) &= \frac{\partial \mathbf{F}_{\text{int}}}{\partial \boldsymbol{u}^{[n]}}\dot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\partial \mathbf{F}_{\text{int}}}{\partial \dot{\boldsymbol{u}}^{[n]}}\ddot{\boldsymbol{u}}^{[n]}(t_n) \\ &= \mathbf{K}_{\text{int}}\dot{\boldsymbol{u}}^{[n]}(t_n) + \mathbf{C}_{\text{int}}\ddot{\boldsymbol{u}}^{[n]}(t_n) \ .\end{aligned} \tag{4.3}$$

Note that all estimators discussed in the previous chapter need to approximate in some way third and successive rates. However, the time-stepping methods only handle vectors of displacements, velocities and accelerations. Then, a question immediately arise. Can we design error estimators in the context of solid and structural dynamics which do not need to approximate higher rates?. The answer is given throughout this chapter.

4.2 General description of the strategy

In Section 3.2 it was concluded that all we need to estimate the local error is to compute an *improved* solution $\tilde{\boldsymbol{y}}_{n+1}^{[n]}$ which is more accurate then the numerical one. Recall that the order of accuracy of this improved solution has to be $\mathcal{O}(\Delta t^{p+2})$, where p is the order of the time-stepping method.

Most of the integration methods commonly used in the context of solid dynamics are second-order accurate (HHT method, generalized-α method and several ones in the Newmark family). For this reason, the methodology presented in this section focuses on this case. Nevertheless, we will briefly discussed in a remark the case of first-order methods.

The general strategy which will be explained next was introduced by Romero & Lacoma [2006c] and it is applicable to both linear and non-linear problems.

Concentrating our development on the second-order case, it has already been mentioned that the order of accuracy of the *improved solution* needs to be $\mathcal{O}(\Delta t^4)$. Let $\boldsymbol{y}^{[n]}(t) = \langle \boldsymbol{u}^{[n]}(t), \dot{\boldsymbol{u}}^{[n]}(t) \rangle$ be the exact solution of the local problem given in (3.3). Then, the value of this exact local solution at t_{n+1} can be expressed via the fundamental theorem of calculus as:

$$\boldsymbol{y}^{[n]}(t_{n+1}) = \underbrace{\boldsymbol{y}^{[n]}(t_n)}_{=\tilde{\boldsymbol{z}}_n} + \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{y}}^{[n]}(\tau)\,\mathrm{d}\tau \ . \tag{4.4}$$

where $\boldsymbol{y}^{[n]}(t_n)$ collects the initial conditions of the local problem which are known and are equal to $\tilde{\boldsymbol{z}}_n$. From the previous equation and as the first term is given, it is deduced that in order to compute the value of $\boldsymbol{y}^{[n]}(t_{n+1})$ it is only required to calculate the integral of $\dot{\boldsymbol{y}}^{[n]}(t)$ in the interval $[t_n, t_{n+1}]$.

Then, by replacing the integral in (4.4) by an approximation of order $\mathcal{O}(\Delta t^4)$, it is straightforward to built an *improved solution* with this same order of accuracy. This approximation to the integral is denoted by $\mathtt{Q}_n^{n+1}$, i.e,

$$\int_{t_n}^{t_{n+1}} \dot{\boldsymbol{y}}^{[n]}(\tau)\,\mathrm{d}\tau = \mathtt{Q}_n^{n+1} + \mathcal{O}(\Delta t^4)\ . \tag{4.5}$$

Thus, once $\mathtt{Q}_n^{n+1}$ is computed, the *improved* solution becomes

$$\tilde{\boldsymbol{y}}_{n+1}^{[n]} = \tilde{\boldsymbol{z}}_n + \mathtt{Q}_n^{n+1}\ , \tag{4.6}$$

and inserting equations (4.5) and (4.6) into (4.4), it is easily deduced that

$$\tilde{\boldsymbol{y}}_{n+1}^{[n]} = \boldsymbol{y}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^4)\ . \tag{4.7}$$

Therefore, recalling the definition of the quantity $\boldsymbol{\eta}_{n+1}$ given in (3.12) and using (4.7), it reads

$$\boldsymbol{\eta}_{n+1} = \boldsymbol{y}^{[n]}(t_{n+1}) - \tilde{\boldsymbol{y}}_{n+1}^{[n]} = \mathcal{O}(\Delta t^4), \tag{4.8}$$

satisfying by construction the condition **C1**.

A possibility to compute $\mathtt{Q}_n^{n+1}$ is to use a three-points Gauss-Lobatto quadrature, which is fourth-order accurate (for more details see, for example, WEISSTEIN [2006]). The choice of this type of quadrature is not arbitrary, since it uses information of the endpoints of the time interval t_n and t_{n+1}, and also of the midpoint $t_{n+1/2}$. As we will show later, the information at t_n and t_{n+1} is known, and therefore, this quadrature requires a low computational cost.

The formula which defines the three-points Gauss-Lobatto quadrature is given by:

$$\mathtt{Q}_n^{n+1} = \frac{\Delta t_n}{2}\left[\frac{1}{3}\dot{\boldsymbol{y}}^{[n]}(t_n) + \frac{4}{3}\dot{\boldsymbol{y}}^{[n]}(t_{n+1/2}) + \frac{1}{3}\dot{\boldsymbol{y}}^{[n]}(t_{n+1})\right]\ . \tag{4.9}$$

From formula (4.9), it follows that to guarantee the fourth-order accuracy of the Gauss-Lobatto quadrature, it is sufficient to know a third-order approximation to the exact values $\dot{\boldsymbol{y}}^{[n]}(t_n)$, $\dot{\boldsymbol{y}}^{[n]}(t_{n+1/2})$ and $\dot{\boldsymbol{y}}^{[n]}(t_{n+1})$ since the terms inside of the parenthesis are multiplied by Δt_n. Then, substituting the exact values $\dot{\boldsymbol{y}}^{[n]}(t_n)$, $\dot{\boldsymbol{y}}^{[n]}(t_{n+1/2})$ and $\dot{\boldsymbol{y}}^{[n]}(t_{n+1})$ by third-order approximations which are denoted by a tilde, the quadrature reads:

$$\mathtt{Q}_n^{n+1} = \frac{\Delta t_n}{2}\left[\frac{1}{3}\dot{\tilde{\boldsymbol{y}}}_n^{[n]} + \frac{4}{3}\dot{\tilde{\boldsymbol{y}}}_{n+1/2}^{[n]} + \frac{1}{3}\dot{\tilde{\boldsymbol{y}}}_{n+1}^{[n]}\right]\ . \tag{4.10}$$

The first term $\dot{\tilde{\boldsymbol{y}}}_n^{[n]}$ collects third-order approximations to exact local velocities and accelerations at instant t_n. The velocity is known exactly since it is the second initial condition (see expression (3.2)) and its value is $\mathbf{V}_n$. Regarding the acceleration, for the moment, we denote by $\tilde{\mathbf{A}}_n$ to be a third-order accurate approximation to $\ddot{\boldsymbol{u}}^{[n]}(t_n)$. Then, the first term becomes

$$\dot{\tilde{\boldsymbol{y}}}_n^{[n]} = \left\{ \begin{array}{c} \mathbf{V}_n \\ \tilde{\mathbf{A}}_n \end{array} \right\} . \tag{4.11}$$

In a second-order method, $\mathbf{V}_{n+1}$ is a third-order approximation to exact velocity $\dot{\boldsymbol{u}}^{[n]}(t_{n+1})$ at the end of the local problem. However, as we will show later, numerical acceleration does not always fulfil this condition, being sometimes less accurate than order $\mathcal{O}(\Delta t^3)$. Hence, using again a correct acceleration $\tilde{\mathbf{A}}_{n+1}$, the third term reads

$$\dot{\tilde{\boldsymbol{y}}}_{n+1}^{[n]} = \left\{ \begin{array}{c} \mathbf{V}_{n+1} \\ \tilde{\mathbf{A}}_{n+1} \end{array} \right\} . \tag{4.12}$$

The second term $\dot{\tilde{\boldsymbol{y}}}_{n+1/2}^{[n]}$ is not as easy as computing as the others two. Let $\mathbf{V}^*_{n+1/2}$ and $\mathbf{A}^*_{n+1/2}$ be third-order approximations to exact velocity and acceleration, respectively, at the midpoint of the time interval. Then, $\dot{\tilde{\boldsymbol{y}}}_{n+1/2}^{[n]}$ is

$$\dot{\tilde{\boldsymbol{y}}}_{n+1/2}^{[n]} = \left\{ \begin{array}{c} \mathbf{V}^*_{n+1/2} \\ \mathbf{A}^*_{n+1/2} \end{array} \right\} . \tag{4.13}$$

These values can be computed by the following set of formulas:

$$\begin{cases} \mathbf{U}^*_{n+1/2} := \mathbf{U}_n + \dfrac{\Delta t_n}{2}\mathbf{V}_n + \dfrac{\Delta t_n^2}{8}\tilde{\mathbf{A}}_n \ , \\ \mathbf{V}^*_{n+1/2} := \mathbf{V}_n + \dfrac{3\Delta t_n}{8}\tilde{\mathbf{A}}_n + \dfrac{\Delta t_n}{8}\tilde{\mathbf{A}}_{n+1} \ , \\ \mathbf{A}^*_{n+1/2} := \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1/2}) - \mathbf{F}_{\text{int}}(\mathbf{U}^*_{n+1/2}, \mathbf{V}^*_{n+1/2})\right] \ . \end{cases} \tag{4.14}$$

It can be easily proved by means of expansions in Taylor's series that formulas given in (4.14) are indeed third-order approximations to exact values at $t_{n+1/2}$.

Finally, having defined all terms to compute the quadrature $\mathtt{Q}_n^{n+1}$, the local error estimate $\boldsymbol{\theta}_{n+1}$ reads

$$\begin{aligned} \boldsymbol{\theta}_{n+1} &:= \tilde{\boldsymbol{y}}_{n+1}^{[n]} - \tilde{\boldsymbol{z}}_{n+1} \\ &= \tilde{\boldsymbol{z}}_n + \mathtt{Q}_n^{n+1} - \tilde{\boldsymbol{z}}_{n+1} \ , \end{aligned} \tag{4.15}$$

where $\tilde{\boldsymbol{z}}_n$ and $\tilde{\boldsymbol{z}}_{n+1}$ are the numerical solution at t_n and t_{n+1}, respectively.

This strategy is directly applicable to different time-stepping methods. The only difference between them is how the accelerations $\tilde{\mathbf{A}}_n$ and $\tilde{\mathbf{A}}_{n+1}$ are computed. In the two following two sections, we particularize this strategy to the Newmark and the HHT method.

Remarks 4.1.

i. For linear problems, the formulation of the new estimator is the same as the one explained. In this case, the equation given in $(4.14)_3$ is particularized to the linear case, and thus, it becomes:

$$\mathbf{A}^*_{n+1/2} := \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1/2}) - \mathbf{C}\mathbf{V}^*_{n+1/2} - \mathbf{K}\mathbf{U}^*_{n+1/2}\right] . \tag{4.16}$$

ii. To compute the acceleration at $t_{n+1/2}$, a linear system of equations has to be solved at each time step. The matrix is the mass matrix $\mathbf{M}$ which is constant along the analysis. It can be factorized at the beginning of the analysis to keep the computational cost low. Another alternative is to use a lumped mass matrix, being trivial to solve the resulting linear system of equations.

iii. When an implicit integration method, such as the HHT, is used to solve a non-linear problem, several Newton-Raphson iterations have to be performed in each time step. In each one of these iterations, a linear system of equations has to be solved. If the computational cost of solving a linear system is $\mathcal{O}(n^3_{\text{dof}})$ and N iterations are needed to converge, then the computational cost of each time step will be $N \cdot \mathcal{O}(n^3_{\text{dof}})$. On the other hand, if a consistent mass matrix is employed to compute the acceleration at $t_{n+1/2}$, the computational cost of solving the system pointed out in **ii** is only $\mathcal{O}(n^2_{\text{dof}})$, being $\mathcal{O}(n^2_{\text{dof}}) \ll N \cdot \mathcal{O}(n^3_{\text{dof}})$. Hence, the cost of computing the error estimator is clearly negligible.

iv. The error estimator introduced in this section has been developed to be employed with second-order methods. If we use a first-order method, then, in principle, the quadrature is not needed to be as accurate as (4.9). One could think of using a Gauss-Lobatto formula with two points. But this quadrature is only second-order accurate, one-order less than neccesary. Thus, we propose to use the same three-points quadrature also for first-order methods. □

4.2.1 A remark about computing the displacement at $t_{n+1/2}$

The approximation used for the displacement at the midpoint (recall equation $(4.14)_1$) only takes into account values of displacement, velocity and acceleration at time t_n. It works correctly when the external force is a continuous function. However, when the force has discontinuities, the approximation defined in $(4.14)_1$ does not provide the required accuracy. To overcome this drawback we propose two different alternatives which use values both of t_n and t_{n+1}. This new displacement at midpoint is denoted by $\tilde{\mathbf{U}}^*_{n+1/2}$ and it reads:

$$\begin{aligned} \tilde{\mathbf{U}}^*_{n+1/2} &:= \frac{3}{4}\mathbf{U}_n + \frac{1}{4}\mathbf{U}_{n+1} + \frac{\Delta t_n}{4}\mathbf{V}_n \ , \qquad \text{or} \\ \tilde{\mathbf{U}}^*_{n+1/2} &:= \frac{1}{2}(\mathbf{U}_n + \mathbf{U}_{n+1}) + \frac{\Delta t_n}{4}(\mathbf{V}_n - \mathbf{V}_{n+1}) + \frac{\Delta t_n^2}{16}(\tilde{\mathbf{A}}_n + \tilde{\mathbf{A}}_{n+1}) \ . \end{aligned} \tag{4.17}$$

Both approximations have the same order of accuracy as $(4.14)_1$ when the external force is a continuous function. In addition, they provide a more realistic value for displacement at midpoint of the interval in the case of existing discontinuous forces.

A typical case of discontinuity in the forces is a contact problem. Next, we illustrate the idea described above in this type of situations. Let us consider a problem of impact of a deformable element against a wall, as depicted in Figure 4.1. From $(4.14)_1$, we note that the approximation used for computing the displacement at the midpoint only takes into account values at time t_n. Therefore, when contact occurs, this expression will compute a wrong value of this displacement, since it will violate the contact condition (see Figure 4.1, position denoted as $\mathbf{U}^*_{n+1/2}$). However, the expressions defined in (4.17) compute a correct value of this displacement (position denoted as $\tilde{\mathbf{U}}^*_{n+1/2}$).

4.3 First case: Newmark's method

In this section we use the general strategy described in Section 4.2 for developing an error estimator for Newmark's method. This new error estimator was introduced in Romero & Lacoma [2006c] in the context of linear problems and later, in Romero & Lacoma [2006a] it was extended to the non-linear case.

Recall that the Newmark family includes first and second-order methods. As mentioned before it is more straightforward to implement the same estimator

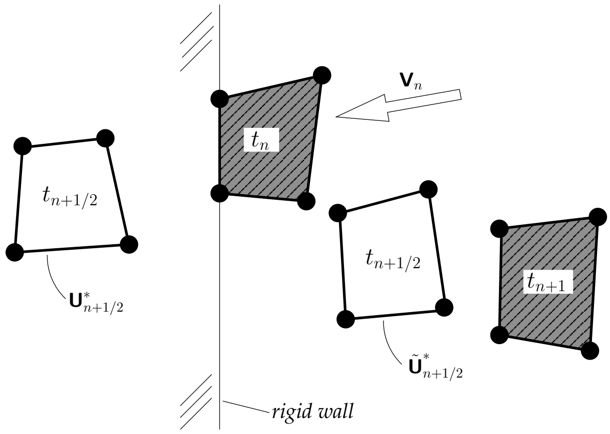


Figure 4.1: Impact of a finite element against a rigid wall. In shaded: position of the element at t_n and t_{n+1}. In white: position of the element at $t_{n+1/2}$ computed using the two approximations introduced. Arrow indicates the direction of velocity.

for both situations. All we need to formulate the local estimate is to compute an acceleration with the correct local order of accuracy. In this case it is very simple since the acceleration provided by the Newmark's method has the same order of accuracy as the displacement and the velocity. For second-order methods the three variables are third-order accurate (see, for example, Hulbert & Hughes [1987]).

Therefore, substituting the accelerations $\tilde{\mathbf{A}}_n$ by $\mathbf{A}_n$ in (4.11), (4.12) and (4.14), an estimate for the local error is immediately obtained.

4.3.1 Summary of the implementation of error estimator

In Box 4.1 we summarize the algorithm to compute the new error estimator for Newmark's method.

Given the solution at t_n, that is, the vectors $\mathbf{U}_n$, $\mathbf{V}_n$, $\mathbf{A}_n$:

1. Compute the solution at t_{n+1}, $\mathbf{U}_{n+1}$, $\mathbf{V}_{n+1}$, $\mathbf{A}_{n+1}$ using the Newmark's method.

2. Compute the displacement, velocity and acceleration at instant $t_{n+1/2}$:

2.a. For displacements and velocities, use the following expressions:

$$
\begin{aligned}
\tilde{\mathbf{U}}^*_{n+1/2} &:= \frac{3}{4}\mathbf{U}_n + \frac{1}{4}\mathbf{U}_{n+1} + \frac{\Delta t_n}{4}\mathbf{V}_n \ , \qquad \text{or} \\
\tilde{\mathbf{U}}^*_{n+1/2} &:= \frac{1}{2}(\mathbf{U}_n + \mathbf{U}_{n+1}) + \frac{\Delta t_n}{4}(\mathbf{V}_n - \mathbf{V}_{n+1}) + \frac{\Delta t_n^2}{16}(\mathbf{A}_n + \mathbf{A}_{n+1}) \ , \\
\mathbf{V}^*_{n+1/2} &:= \mathbf{V}_n + \frac{3\Delta t_n}{8}\mathbf{A}_n + \frac{\Delta t_n}{8}\mathbf{A}_{n+1} \ .
\end{aligned}
\tag{4.18}
$$

2.b. For the acceleration, first compute the vector of internal forces at $t_{n+1/2}$ and, subsequently, solve the following linear system of equations:

$$
\mathbf{M}\mathbf{A}^*_{n+1/2} = \mathbf{F}(t_{n+1/2}) - \mathbf{F}_{\text{int}}(\tilde{\mathbf{U}}^*_{n+1/2}, \mathbf{V}^*_{n+1/2}) \ . \tag{4.19}
$$

3. Compute the quadrature rule

$$
\mathsf{Q}_n^{n+1} = \frac{\Delta t_n}{6}\left\{ \begin{array}{c} \mathbf{V}_n \\ \mathbf{A}_n \end{array} \right\} + \frac{4\Delta t_n}{6}\left\{ \begin{array}{c} \mathbf{V}^*_{n+1/2} \\ \mathbf{A}^*_{n+1/2} \end{array} \right\} + \frac{\Delta t_n}{6}\left\{ \begin{array}{c} \mathbf{V}_{n+1} \\ \mathbf{A}_{n+1} \end{array} \right\} \ . \tag{4.20}
$$

4. Compute the error estimation and its energy norm

$$
\boldsymbol{\theta}_{n+1} = \tilde{\boldsymbol{z}}_n + \mathsf{Q}_n^{n+1} - \tilde{\boldsymbol{z}}_{n+1} \implies ||\boldsymbol{\theta}_{n+1}||_E \ . \tag{4.21}
$$

5. Set $n \leftarrow n+1$:

$$
\begin{aligned}
&\texttt{IF } t_n < T \texttt{ THEN GO TO } \mathbf{1}. \\
&\texttt{IF } t_n = T \texttt{ EXIT}
\end{aligned}
\tag{4.22}
$$

Box 4.1: Summary of the implementation of the local error estimate for Newmark's method

4.4 Second case: HHT method

The HHT method is a second-order accurate method HILBER ET AL. [1977] and, thus, it can be proven that the algorithmic displacements and velocities fulfil, respectively, the following relations:

$$\mathbf{U}_{n+1} = \boldsymbol{u}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3), \qquad \mathbf{V}_{n+1} = \dot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \ . \tag{4.23}$$

That is, algorithmic displacements and velocities are third-order approximations to the exact displacements and velocities values at instant t_{n+1} of the local problem (HULBERT & HUGHES [1987]). However, the algorithmic acceleration is only a first-order approximation to the exact local acceleration at t_{n+1}. This result is proved in an appendix at the end of the chapter.

Another typical time-stepping method, the generalized-α method proposed in CHUNG & HULBERT [1993] also presents a lower order of approximation in the accelerations than in displacements and velocities. This aspect is pointed out in ERLICHER ET AL. [2002].

Therefore, to use the strategy presented in Section 4.2, we have to compute a sufficiently accurate acceleration. This is developed in the next section.

4.4.1 Computing an acceleration with the correct local order

As mentioned in Section 4.2, for a correct formulation of the new error estimator in a second-order integration method, the three unknowns displacements, velocities, and accelerations must be third order approximations to the exact local ones. For this reason, the acceleration which the HHT method computes cannot be used to calculate the new error estimator and a new acceleration must be recomputed.

In LACOMA & ROMERO [2006a], two different ways of computing this new acceleration are proposed:

1. The first one is based on recovering the acceleration from the equilibrium equation at instant t_{n+1}, i.e.,

$$\tilde{\mathbf{A}}^{\mathrm{HHT}}_{n+1} = \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1}) - \mathbf{F}_{\mathrm{int}}(\mathbf{U}_{n+1}, \mathbf{V}_{n+1})\right] \ . \tag{4.24}$$

This strategy needs to solve a linear system of equations in each time step. Since mass matrix $\mathbf{M}$ is constant, it only has to be factorized once at the beginning of the analysis.

2. The second alternative is more efficient computationally. This new acceleration is computed from the following recurrence equation

$$\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} = \frac{1}{\alpha}\left[\mathbf{A}_{n+1} - (1-\alpha)\tilde{\mathbf{A}}_{n}^{\mathrm{HHT}}\right] , \qquad \text{with } \tilde{\mathbf{A}}_{0}^{\mathrm{HHT}} = \mathbf{A}_0 , \tag{4.25}$$

where $\mathbf{A}_{n+1}$ is the acceleration computed by the HHT method and α is the factor which defines this method. In Proposition 4.2, it will be proven that this new acceleration is indeed a third order approximation to the exact acceleration at t_{n+1} of the local problem.

Before that, we present a useful result which studies the order of approximation of the algorithmic acceleration to the exact local value at $t_{n+\alpha}$.

Proposition 4.1. *The acceleration computed by the HHT method and denoted $\mathbf{A}_{n+1}$ is a second order approximation to the exact acceleration at time $t_{n+\alpha}$, $\ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})$, of the local problem defined in (3.2), i.e.,*

$$\mathbf{A}_{n+1} = \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha}) + \underbrace{\frac{\Delta t_n^2}{2}\alpha(1-\alpha)\ddddot{\boldsymbol{u}}^{[n]}(t_n) + \mathcal{O}(\Delta t^3)}_{\mathcal{O}(\Delta t^2)} , \tag{4.26}$$

Proof: In order to prove this proposition, and to make calculations easier, the difference $\mathbf{M}\Big[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\Big]$ is studied. Using the balance equation, this difference yields

$$\begin{aligned}\mathbf{M}\Big[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\Big] = {} & \mathbf{F}_{n+\alpha} - (1-\alpha)\mathbf{C}\mathbf{V}_n - \alpha\mathbf{C}\mathbf{V}_{n+1} - (1-\alpha)\mathbf{K}\mathbf{U}_n \\ & - \alpha\mathbf{K}\mathbf{U}_{n+1} - \mathbf{M}\ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha}) .\end{aligned} \tag{4.27}$$

The first step consists of inserting the expressions defined into (4.23) into equation (4.27). Then, the exact solution of the local problem $\boldsymbol{u}^{[n]}(t_{n+1})$ and $\dot{\boldsymbol{u}}^{[n]}(t_{n+1})$, are expanded by Taylor series about t_n, and taking into account the initial conditions $(3.2)_2$ and $(3.2)_3$, it follows

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\right] = \mathbf{F}_{n+\alpha} &- (1-\alpha)\left[\mathbf{C}\mathbf{V}_n + \mathbf{K}\mathbf{U}_n\right] - \mathbf{M}\ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha}) \\ &- \alpha\mathbf{C}\left[\mathbf{V}_n + \Delta t_n \ddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\Delta t_n^2}{2}\dddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] \\ &- \alpha\mathbf{K}\left[\mathbf{U}_n + \Delta t_n \mathbf{V}_n + \frac{\Delta t_n^2}{2}\ddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] + \mathcal{O}(\Delta t^3)\ .\end{aligned} \tag{4.28}$$

Simplifying terms and expanding the exact acceleration $\ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})$ by Taylor series about t_n up to order two, the following expression is reached:

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\right] = \mathbf{F}_{n+\alpha} &- \mathbf{C}\mathbf{V}_n - \mathbf{K}\mathbf{U}_n \\ &- \alpha\mathbf{C}\left[\Delta t_n \ddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\Delta t_n^2}{2}\dddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] \\ &- \alpha\mathbf{K}\left[\Delta t_n \mathbf{V}_n + \frac{\Delta t_n^2}{2}\ddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] \\ &- \mathbf{M}\left[\ddot{\boldsymbol{u}}^{[n]}(t_n) + \alpha\Delta t_n \dddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\alpha^2\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[n]}(t_n)\right] + \mathcal{O}(\Delta t^3)\ .\end{aligned} \tag{4.29}$$

Using the balance equation $(3.2)_1$ at time t_n and reordering terms yields

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\right] = \mathbf{F}_{n+\alpha} &- \mathbf{F}(t_n^+) - \alpha\Delta t_n\left[\mathbf{M}\dddot{\boldsymbol{u}}^{[n]}(t_n) + \mathbf{C}\ddot{\boldsymbol{u}}^{[n]}(t_n) + \mathbf{K}\mathbf{V}_n\right] \\ &- \frac{\alpha\Delta t_n^2}{2}\left[\alpha\mathbf{M}\ddddot{\boldsymbol{u}}^{[n]}(t_n) + \mathbf{C}\dddot{\boldsymbol{u}}^{[n]}(t_n) + \mathbf{K}\ddot{\boldsymbol{u}}^{[n]}(t_n)\right] + \mathcal{O}(\Delta t^3)\ .\end{aligned} \tag{4.30}$$

Now, expressing $\mathbf{F}_{n+\alpha}$ in terms of $\mathbf{F}(t_n^+)$ and $\mathbf{F}(t_{n+1}^-)$ via (2.27) and taking into account the first and the second time derivative of the balance equation $(3.2)_1$, the expression (4.30) is reduced to

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha})\right] &= \alpha\left[\mathbf{F}(t_{n+1}^-) - \mathbf{F}(t_n^+)\right] - \alpha\Delta t_n\dot{\mathbf{F}}(t_n^+) - \frac{\alpha\Delta t_n^2}{2}\ddot{\mathbf{F}}(t_n^+) \\ &\quad + \alpha(1-\alpha)\frac{\Delta t_n^2}{2}\mathbf{M}\ddddot{\boldsymbol{u}}^{[n]}(t_n) + \mathcal{O}(\Delta t^3) \\ &= \alpha(1-\alpha)\frac{\Delta t_n^2}{2}\mathbf{M}\ddddot{\boldsymbol{u}}^{[n]}(t_n) \\ &\quad + \underbrace{\alpha\left[\mathbf{F}_{n+1^-} - \mathbf{F}(t_n^+) - \Delta t_n\dot{\mathbf{F}}(t_n^+) - \frac{\Delta t_n^2}{2}\ddot{\mathbf{F}}(t_n^+)\right]}_{\mathcal{O}(\Delta t^3)} + \mathcal{O}(\Delta t^3)\ .\end{aligned} \tag{4.31}$$

Thus,

$$\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha}) = \alpha(1-\alpha)\frac{\Delta t_n^2}{2}\dddot{\boldsymbol{u}}^{[n]}(t_n) + \mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^2) \tag{4.32}$$

■

Remarks 4.2.

i. In this proposition we are studying the local problem defined in the time interval $[t_n, t_{n+1}]$. Thus, in expressions (4.30) and (4.31), the external force and its time derivatives at t_n and t_{n+1} must be substituted, respectively, by the limits from the right and the left in order to avoid discontinuities at the endpoints of the interval.

ii. In expression (4.32), if $\alpha = 1$, i.e., the Newmark's method is selected, the first term vanishes and thus, the algorithmic acceleration is a third order approximation to the local exact acceleration at equilibrium point t_{n+1}.

□

Finally, the recurrence law defined in (4.25) is proven.

Proposition 4.2. *In the HHT method, the new acceleration* $\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}}$ *computed by (4.25) is a third order approximation to the exact acceleration,* $\ddot{\boldsymbol{u}}^{[n]}(t_{n+1})$*, of the local problem defined in (3.2). This order of approximation holds if and only if the function that defines the external forces is continuous.*

PROOF: This proposition is proven by induction. Firstly, the difference $\tilde{\mathbf{A}}_1^{\mathrm{HHT}} - \ddot{\boldsymbol{u}}^{[0]}(t_1)$ obtained in the first integration step is investigated. For this purpose, it is necessary to remark that $\mathbf{A}_0 = \ddot{\boldsymbol{u}}^{[0]}(t_0)$. Moreover, the result presented in

Proposition 4.1 is employed.

$$
\begin{aligned}
\tilde{\mathbf{A}}_1^{\mathrm{HHT}} - \ddot{\boldsymbol{u}}^{[0]}(t_1) &= \frac{1}{\alpha}\left[\mathbf{A}_1 - (1-\alpha)\mathbf{A}_0\right] - \ddot{\boldsymbol{u}}^{[0]}(t_1) \\
&= \frac{1}{\alpha}\left[\ddot{\boldsymbol{u}}^{[0]}(t_\alpha) + \frac{\Delta t_n^2}{2}\alpha(1-\alpha)\ddddot{\boldsymbol{u}}^{[0]}(t_0) - (1-\alpha)\ddot{\boldsymbol{u}}^{[0]}(t_0)\right] \\
&\quad - \ddot{\boldsymbol{u}}^{[0]}(t_1) + \mathcal{O}(\Delta t^3) \\
&= \frac{1}{\alpha}\left[\ddot{\boldsymbol{u}}^{[0]}(t_0) + \alpha\Delta t_n \dddot{\boldsymbol{u}}^{[0]}(t_0) + \frac{\alpha^2\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[0]}(t_0)\right. \\
&\quad \left. + \frac{\Delta t_n^2}{2}\alpha(1-\alpha)\ddddot{\boldsymbol{u}}^{[0]}(t_0) - (1-\alpha)\ddot{\boldsymbol{u}}^{[0]}(t_0)\right] - \ddot{\boldsymbol{u}}^{[0]}(t_1) + \mathcal{O}(\Delta t^3) \\
&= \ddot{\boldsymbol{u}}^{[0]}(t_0) + \Delta t_n \dddot{\boldsymbol{u}}^{[0]}(t_0) + \frac{\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[0]}(t_0) - \ddot{\boldsymbol{u}}^{[0]}(t_1) + \mathcal{O}(\Delta t^3) \\
&= \ddot{\boldsymbol{u}}^{[0]}(t_1) - \ddot{\boldsymbol{u}}^{[0]}(t_1) + \mathcal{O}(\Delta t^3) \\
&= \mathcal{O}(\Delta t^3)
\end{aligned}
\tag{4.33}
$$

Now, it must be proven that if $\tilde{\mathbf{A}}_n^{\mathrm{HHT}} = \ddot{\boldsymbol{u}}^{[n-1]}(t_n) + \mathcal{O}(\Delta t^3)$ this relation also holds a time step after, i.e., at t_{n+1}. The process followed in this case is very similar to the one just presented in (4.33) but here the approximation introduced in Lemma 3.1 is taken into account.

$$
\begin{aligned}
\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) &= \frac{1}{\alpha}\left[\mathbf{A}_{n+1} - (1-\alpha)\mathbf{A}_n\right] - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) \\
&= \frac{1}{\alpha}\left[\ddot{\boldsymbol{u}}^{[n]}(t_{n+\alpha}) + \frac{\Delta t_n^2}{2}\alpha(1-\alpha)\ddddot{\boldsymbol{u}}^{[n]}(t_n) - (1-\alpha)\ddot{\boldsymbol{u}}^{[n-1]}(t_n)\right] \\
&\quad - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \\
&= \frac{1}{\alpha}\left[\ddot{\boldsymbol{u}}^{[n]}(t_n) + \alpha\Delta t_n \dddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\alpha^2\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[n]}(t_n)\right. \\
&\quad \left. + \frac{\Delta t_n^2}{2}\alpha(1-\alpha)\ddddot{\boldsymbol{u}}^{[n]}(t_n) - (1-\alpha)\ddot{\boldsymbol{u}}^{[n]}(t_n)\right] - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \\
&= \ddot{\boldsymbol{u}}^{[n]}(t_n) + \Delta t_n \dddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\Delta t_n^2}{2}\ddddot{\boldsymbol{u}}^{[n]}(t_n) - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \\
&= \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t^3) \\
&= \mathcal{O}(\Delta t^3)
\end{aligned}
\tag{4.34}
$$

■

4.4.2 Study of the order of convergence of accelerations in a system with one degree of freedom

The objective of this example is to emphasize some of the results presented in Section 4.4.1. For this, we shall investigate the order of convergence of the acceleration provided by the HHT method and the new one introduced in Proposition 4.2.

The problem analyzed is a linear mass-spring system with damping and with one degree of freedom. The value of the mass is 1 kg, the stiffness 1 N/m and the damping 1 Ns/m. The system is excited with a linear load proportional to the time, with proportionality constant 1 N/s. With these data the problem is defined as follows:

$$\begin{cases} \ddot{\mathbf{U}}(t) + \dot{\mathbf{U}}(t) + \mathbf{U}(t) = t, \quad t \in [0,1] \ , \\ \mathbf{U}(0) = 0 \ , \\ \dot{\mathbf{U}}(0) = 0 \ . \end{cases} \tag{4.35}$$

This problem is solved using the HHT method with two different values for the parameter α. In the first case, we use a dissipative scheme taking $\alpha = 0.9$, whereas in the second one, a non-dissipative method is employed. For that, we select $\alpha = 1.0$ resulting in the well-known trapezoidal rule.

In this problem the exact solution and thus, the exact acceleration can be easily computed. For this reason, it is feasible to compare the acceleration provided by the HHT method and the new one obtained using the formula (4.25) with the exact acceleration of the problem.

In Figure 4.2, the local order of convergence of accelerations at time $t = 1$ s is shown for both values of parameter α.

First, we study the case in which $\alpha = 0.9$. In Proposition 4.1, we pointed out that the acceleration computed by the HHT method, $\mathbf{A}_{n+1}$, is a second order approximation to the exact local acceleration at instant $t_{n+\alpha}$. This fact can be also deduced from Figure 4.2 (line with "squares"). However, $\mathbf{A}_{n+1}$ is only a first order approximation to the exact local acceleration at t_{n+1} (line with "triangles"). On the other hand, the new acceleration $\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}}$ is indeed more accurate, being a third order approximation (line with "circles") and, thus, being useful to formulate the error estimator proposed in Section 4.2.

When the parameter α is equal to 1.0 (trapezoidal rule), the acceleration provided by the HHT method and the new one computed using the expression (4.25) are the same. They are both a third order approximation to the exact acceleration of the local problem (see Figure 4.2 on the right).

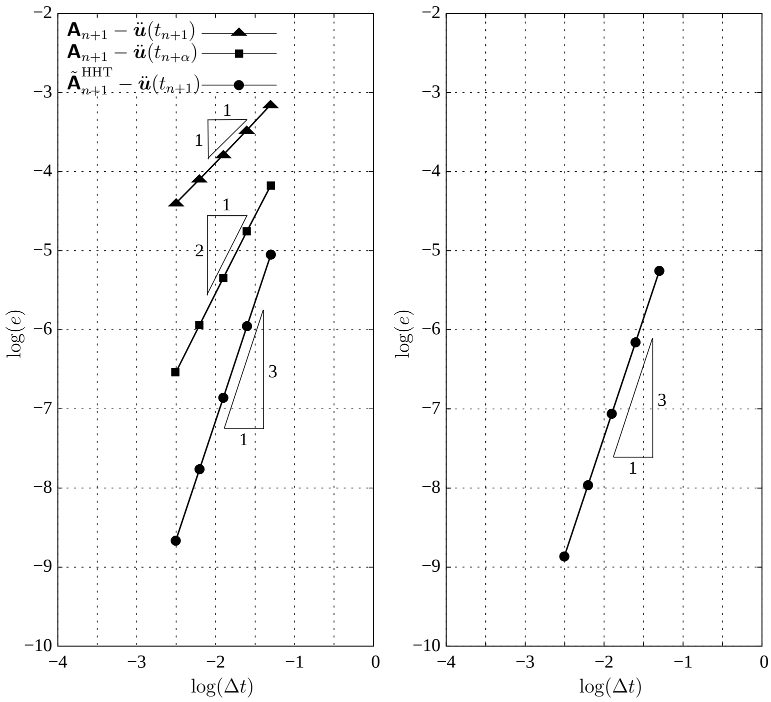


Figure 4.2: Comparative of the local order of convergence of accelerations in the HHT method at $t = 1$ s (Left: HHT method with $\alpha = 0.9$. Right: HHT method with $\alpha = 1.0$)

4.4.3 Implementation of the error estimator for the HHT method

Once a new acceleration with a sufficient accuracy is obtained, the quadrature $\mathtt{Q}_n^{n+1}$ defined in (4.9) can be computed. The first and third term in (4.9) are calculated, respectively, in the following way:

$$\dot{\boldsymbol{y}}_n^{[n]} = \left\{ \begin{array}{c} \mathbf{V}_n \\ \tilde{\mathbf{A}}_n^{\mathrm{HHT}} \end{array} \right\} , \tag{4.36}$$

and

$$\dot{\boldsymbol{y}}_{n+1}^{[n]} = \left\{ \begin{array}{c} \mathbf{V}_{n+1} \\ \tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} \end{array} \right\} , \tag{4.37}$$

which are indeed third-order approximations to the exact values.

In order to obtain the second term of the quadrature in an efficient way, we define algorithmic values for displacement, velocity and acceleration at the midpoint of the time interval using known values of these variables at t_n and t_{n+1}. This strategy is the same as that followed in Section 4.2. Then, the values at midpoint are:

$$\left\{\begin{array}{l} \tilde{\mathbf{U}}^*_{n+1/2} := \dfrac{3}{4}\mathbf{U}_n + \dfrac{1}{4}\mathbf{U}_{n+1} + \dfrac{\Delta t_n}{4}\mathbf{V}_n \ , \qquad \text{or} \\ \tilde{\mathbf{U}}^*_{n+1/2} := \dfrac{1}{2}(\mathbf{U}_n + \mathbf{U}_{n+1}) + \dfrac{\Delta t_n}{4}(\mathbf{V}_n - \mathbf{V}_{n+1}) + \dfrac{\Delta t_n^2}{16}(\tilde{\mathbf{A}}_n^{\mathrm{HHT}} + \tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}}) \ , \\ \mathbf{V}^*_{n+1/2} := \mathbf{V}_n + \dfrac{3\Delta t_n}{8}\tilde{\mathbf{A}}_n^{\mathrm{HHT}} + \dfrac{\Delta t_n}{8}\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} \ , \\ \mathbf{A}^*_{n+1/2} := \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1/2}) - \mathbf{F}_{\mathrm{int}}(\tilde{\mathbf{U}}^*_{n+1/2}, \mathbf{V}^*_{n+1/2})\right] \ . \end{array}\right. \tag{4.38}$$

Note that the new acceleration $\tilde{\mathbf{A}}_n^{\mathrm{HHT}}$ has to be used instead of $\mathbf{A}_n$ to provide values with a correct order of approximation.

With expressions given in (4.38), the middle term in quadrature (4.10) yields

$$\begin{aligned} \dot{\boldsymbol{y}}^{[n]}_{n+1/2} &= \left\{ \begin{array}{c} \mathbf{V}^*_{n+1/2} \\ \mathbf{A}^*_{n+1/2} \end{array} \right\} \\ &= \left\{ \begin{array}{c} \mathbf{V}_n + \dfrac{3\Delta t_n}{8}\tilde{\mathbf{A}}_n^{\mathrm{HHT}} + \dfrac{\Delta t_n}{8}\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} \\ \mathbf{M}^{-1}\left[\mathbf{F}(t_{n+1/2}) - \mathbf{F}_{\mathrm{int}}(\tilde{\mathbf{U}}^*_{n+1/2}, \mathbf{V}^*_{n+1/2})\right] \end{array} \right\} \ . \end{aligned} \tag{4.39}$$

4.4.4 Summary of the implementation of error estimator

The algorithm to compute the new error estimator for the HHT method is summarized in Box 4.2.

Given the solution at t_n, that is, the vectors $\mathbf{U}_n$, $\mathbf{V}_n$, $\mathbf{A}_n$:

1. Compute the solution at t_{n+1}, $\mathbf{U}_{n+1}$, $\mathbf{V}_{n+1}$, $\mathbf{A}_{n+1}$ using the HHT method.

2. Compute the new acceleration at t_{n+1}:

$$\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} = \frac{1}{\alpha}[\mathbf{A}_{n+1} - (1-\alpha)\tilde{\mathbf{A}}_n^{\mathrm{HHT}}] \ . \tag{4.40}$$

3. Compute the displacement, velocity and acceleration at instant $t_{n+1/2}$:

3.a. For displacements and velocities, use the following expressions:

$$\begin{aligned}
\tilde{\mathbf{U}}_{n+1/2}^{*} &:= \frac{3}{4}\mathbf{U}_n + \frac{1}{4}\mathbf{U}_{n+1} + \frac{\Delta t_n}{4}\mathbf{V}_n \ , \qquad \text{or} \\
\tilde{\mathbf{U}}_{n+1/2}^{*} &:= \frac{1}{2}(\mathbf{U}_n + \mathbf{U}_{n+1}) + \frac{\Delta t_n}{4}(\mathbf{V}_n - \mathbf{V}_{n+1}) + \frac{\Delta t_n^2}{16}(\tilde{\mathbf{A}}_n^{\mathrm{HHT}} + \tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}}) \ , \\
\mathbf{V}_{n+1/2}^{*} &:= \mathbf{V}_n + \frac{3\Delta t_n}{8}\tilde{\mathbf{A}}_n^{\mathrm{HHT}} + \frac{\Delta t_n}{8}\tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}} \ .
\end{aligned} \tag{4.41}$$

3.b. For the acceleration, first compute the vector of internal forces at $t_{n+1/2}$ and, subsequently, solve the following linear system of equations:

$$\mathbf{M}\mathbf{A}_{n+1/2}^{*} = \mathbf{F}(t_{n+1/2}) - \mathbf{F}_{\mathrm{int}}(\tilde{\mathbf{U}}_{n+1/2}^{*}, \mathbf{V}_{n+1/2}^{*}) \ . \tag{4.42}$$

4. Compute the quadrature rule:

$$\mathsf{Q}_n^{n+1} = \frac{\Delta t_n}{6}\left\{\begin{array}{c}\mathbf{V}_n \\ \tilde{\mathbf{A}}_n^{\mathrm{HHT}}\end{array}\right\} + \frac{4\Delta t_n}{6}\left\{\begin{array}{c}\mathbf{V}_{n+1/2}^{*} \\ \mathbf{A}_{n+1/2}^{*}\end{array}\right\} + \frac{\Delta t_n}{6}\left\{\begin{array}{c}\mathbf{V}_{n+1} \\ \tilde{\mathbf{A}}_{n+1}^{\mathrm{HHT}}\end{array}\right\} \ . \tag{4.43}$$

5. Compute the error estimation and its energy norm:

$$\boldsymbol{\theta}_{n+1} = \tilde{\boldsymbol{z}}_n + \mathsf{Q}_n^{n+1} - \tilde{\boldsymbol{z}}_{n+1} \implies ||\boldsymbol{\theta}_{n+1}||_E \ . \tag{4.44}$$

6. Set $n \leftarrow n+1$:

$$\begin{aligned}
&\texttt{IF } t_n < T \texttt{ THEN GO TO } \mathbf{1}. \\
&\texttt{IF } t_n = T \texttt{ EXIT}
\end{aligned} \tag{4.45}$$

Box 4.2: Summary of the implementation of the local error estimate for the HHT method

4.5 Representative numerical simulations

The objective of this section is to investigate the performance of the new estimator developed in this chapter. For this, we analyze its behavior throughout six different simulations.

The first three simulations are linear and they are performed employing different schemes of the Newmark family. In these examples, estimates provided by the new estimator are compared to the results computed by the estimator of Wiberg & Li [1993] and the one of Hulbert & Jang [1995]. This last estimator is very similar to the proposed by Zienkiewicz & Xie [1991] being the difference in that it provides an error estimate for velocities. This new term is

$$\boldsymbol{\theta}^{v}_{n+1} = \Delta t_n \left(\gamma - \frac{1}{2}\right) \left(\mathbf{A}_{n+1} - \mathbf{A}_n\right) , \tag{4.46}$$

and together with the error in displacements given in (3.30), they allow us to compute an estimate in the energy norm.

The last three examples are more complex problems. They are non-linear simulations and we use the HHT method to solve them. In these cases, the new estimator will be compared to "exact" errors which are obtained by solving the problem with a much smaller time step and assuming this reference solution as "exact".

4.5.1 One degree of freedom spring-mass system

The system studied in this first simulation is the same as in Section 3.7. It is a mass-spring system of one degree of freedom and with the following characteristics: stiffness $\mathbf{K} = 6$ N/m, mass $\mathbf{M} = 1$ kg and initial displacement $\mathbf{U}(0) = 1$ m. Recall that differential equation of this system was given in (3.53).

Now, a more extensive study is carried out. First, the mass is excited with a sinusoidal load and a first-order method is used to integrate the response of the system. Second, this same analysis is made but employing a second-order method. Finally, a triangular load is applied and, again, a second-order method is used.

In all cases, we compare the results provided by error estimators of Wiberg & Li [1993] and Hulbert & Jang [1995] to that one computed with the new estimator developed in Sections 4.2 and 4.3.

Sinusoidal load and first-order method

In the first case, we employ a first-order and dissipative method within the Newmark family. The parameters selected are: $\beta = 0.36$ and $\gamma = 0.7$. The following smooth load is applied to the mass:

$$\mathsf{F}(t) = \sin(2\pi t) \ , \qquad t \in [0, 5] \ . \tag{4.47}$$

Figure 4.3 depicts the rate of convergence of the local error estimates measured in the energy norm. Both the error estimator of HULBERT & JANG [1995], WIBERG & LI [1993] and the new one provide similar results. Also, the local exact error is represented, and it is very similar to the others three. As a first-order accurate method is being used for the time integration, the local error estimates must be one-order more accurate, i.e., exhibit a second-order accuracy. From Figure 4.3 it can be seen that all the estimators compared have a satisfactory rate of convergence.

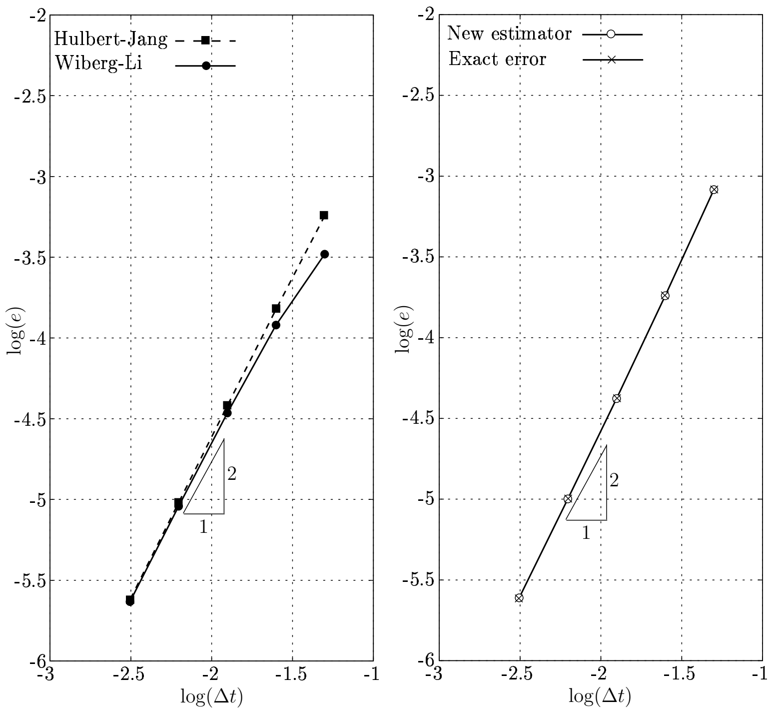


Figure 4.3: One d.o.f. system. Sinusoidal loading. Energy norm of the estimated local errors at time $t = 5.0$ s (Newmark's method with $\beta = 0.36$ and $\gamma = 0.7$).

As discussed in Section 3.3, it is not sufficient with that the local estimation $\boldsymbol{\theta}_{n+1}$ is of size $\mathcal{O}(\Delta t^{p+1})$, in this case $\mathcal{O}(\Delta t^2)$. Besides this, the quantity $\boldsymbol{\eta}_{n+1}$

must be of one order more, so that it can be neglected when it is compared to $\boldsymbol{\theta}_{n+1}$. To study this, we plot in Figure 4.4 the quantity $\boldsymbol{\eta}_{n+1}$ for different time steps. The three estimators analyzed verify this condition. Note that the new estimator is slightly more accurate than the others, since it has been developed for second-order methods.

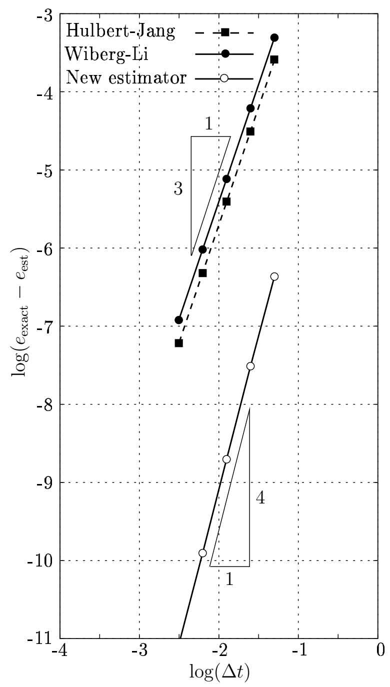


Figure 4.4: One d.o.f. system. Sinusoidal loading. Difference between estimated and exact local errors, measured in the energy norm, at time $t = 5.0$ s (Newmark's method with $\beta = 0.36$ and $\gamma = 0.7$).

Figure 4.5 depicts the evolution of the estimated local errors using a constant time step $\Delta t = 0.05$ s. In this simple problem, it is feasible to solve exactly each local problem, obtaining, therefore, the exact local error. By comparing the four graphics, it is seen that the three estimators compute local estimates similar between them and they are very close to the exact one.

Adding up the local contributions as explained in Theorem 3.2, an estimate for the global error is obtained. It is plotted in Figure 4.6. Again, the estimate global errors provided by three estimators are very close and they are an accurate upper bound for the true error.

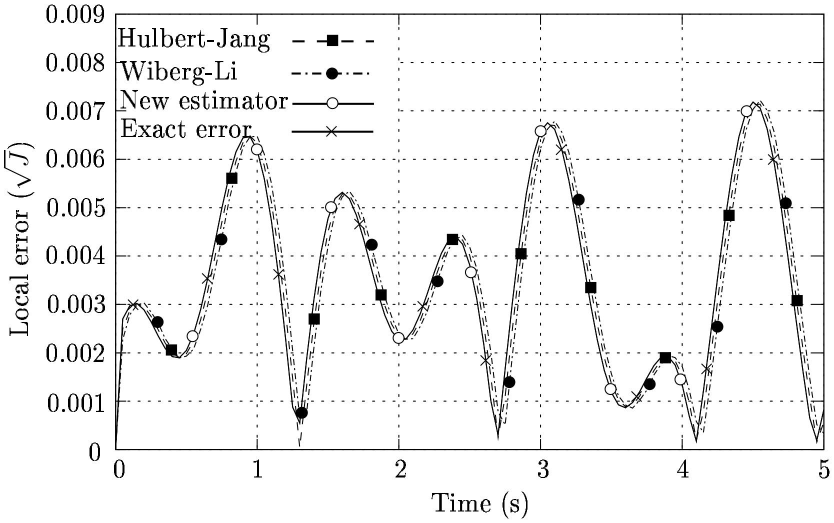


Figure 4.5: One d.o.f. system. Sinusoidal loading. Local error in energy norm with time step $\Delta t = 0.05$ s (Newmark's method with $\beta = 0.36$ and $\gamma = 0.7$).

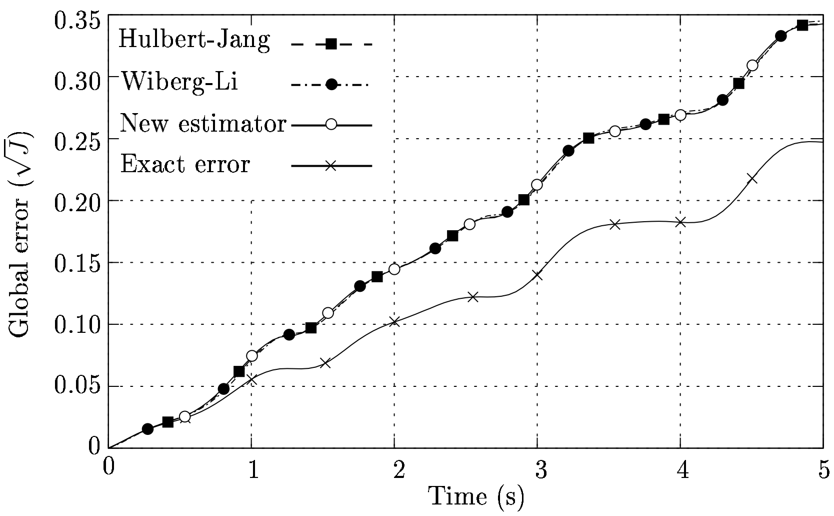


Figure 4.6: One d.o.f. system. Sinusoidal loading. Global error in energy norm with time step $\Delta t = 0.05$ s (Newmark's method with $\beta = 0.36$ and $\gamma = 0.7$).

Figure 4.7 shows the rate of convergence for the estimated global errors obtained with the three estimators compared and also the exact one is included. It is desirable that the global error predicted by an estimator converge at the same rate as the true error in order to compute accurate estimates. For the problem under consideration, first-order rate of convergence is confirmed for all the error estimators.

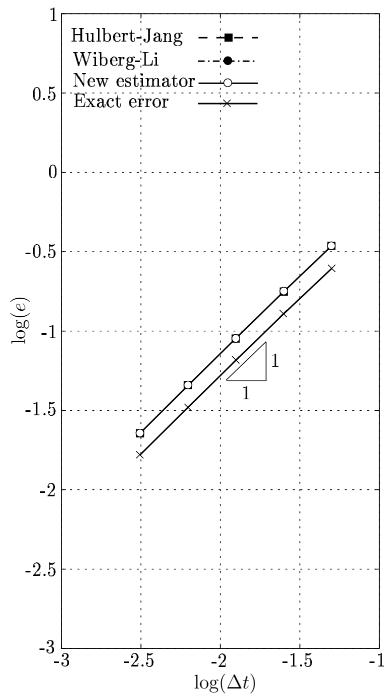


Figure 4.7: One d.o.f. system. Sinusoidal loading. Estimated global errors at time $t = 5.0$ s (Newmark's method with $\beta = 0.36$ and $\gamma = 0.7$).

As final commentary to this first case, we conclude that the three estimators exhibit a good performance since they compute errors very close to the exact values.

Sinusoidal load and second-order method

In the second simulation we discuss the numerical solution of the same spring-mass system excited, again, with the sinusoidal load given in (4.47), but using now as integration scheme the trapezoidal rule, a second order method of the Newmark family.

In contrast to previous simulation, in this case, the estimators analyzed start to exhibit differences in their behaviors.

Firstly, the estimated local errors as functions of the time step are compared in Figure 4.8. For the integrator employed, local error estimates must be of size $\mathcal{O}(\Delta t^3)$, which is confirmed for all the estimators considered by the slopes of the lines of this figure. The values obtained by the new estimator are superimposed over the exact values, confirming its good accuracy. Also, we point out that the estimator of HULBERT & JANG [1995] underestimates the local error, although it will be better observed in the next figures.

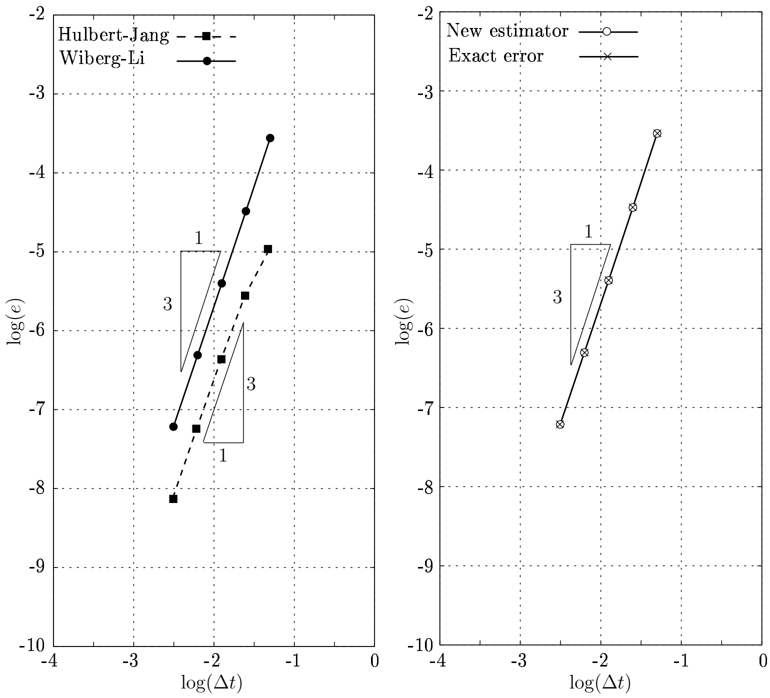


Figure 4.8: One d.o.f. system. Sinusoidal loading. Estimated local errors at time $t = 5.0$ s (trapezoidal rule).

Secondly, the unresolved part of the error is compared in Figure 4.9. For the trapezoidal rule, this quantity must be, at least, of size $\mathcal{O}(\Delta t^4)$. Both the new estimator and the estimator of WIBERG & LI [1993] satisfy this condition. However, the estimator of HULBERT & JANG [1995] fails to comply with this condition, exhibiting an unresolved error one order less than needed. This fact causes this estimator to neglect a important part of the exact local error, underestimating it. Recall that this topic was explained in Sections 3.3 and 3.4.1.

Third, the evolution of the local estimated errors for a fixed time step $\Delta t = 0.05$ s is depicted in Figure 4.10. The plot shows very good agreement between

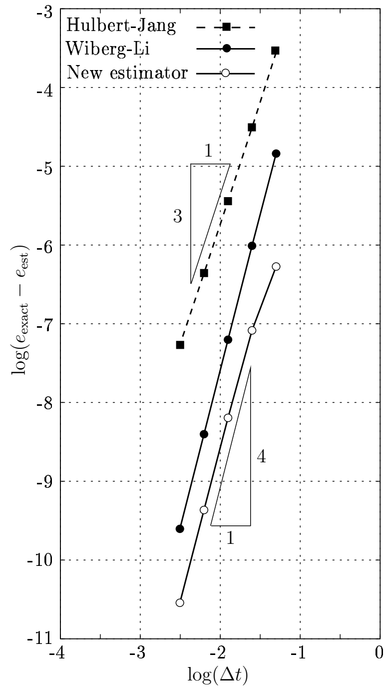


Figure 4.9: One d.o.f. system. Sinusoidal loading. Difference between estimated and exact local errors at time $t = 5.0$ s (trapezoidal rule).

the exact local errors and the estimations provided by the method of WIBERG & LI [1993] and the newly proposed one. The accuracy of the estimator of HULBERT & JANG [1995] is not so good. In this figure and in the next one, it is clearly observed that this estimator does not provide safe bounds for the error.

Keeping the same time step as before, the evolution of the exact and estimated global errors are shown in Figure 4.11. As deduced in Section 3.4.1, recall that estimator of HULBERT & JANG [1995] fails to satisfy the condition **C1**. Therefore, the use of this estimator is dangerous since we cannot ensure that the error estimation is reliable. However, in smooth problems like this one, the estimator of WIBERG & LI [1993] and the new one are accurate and reliable.

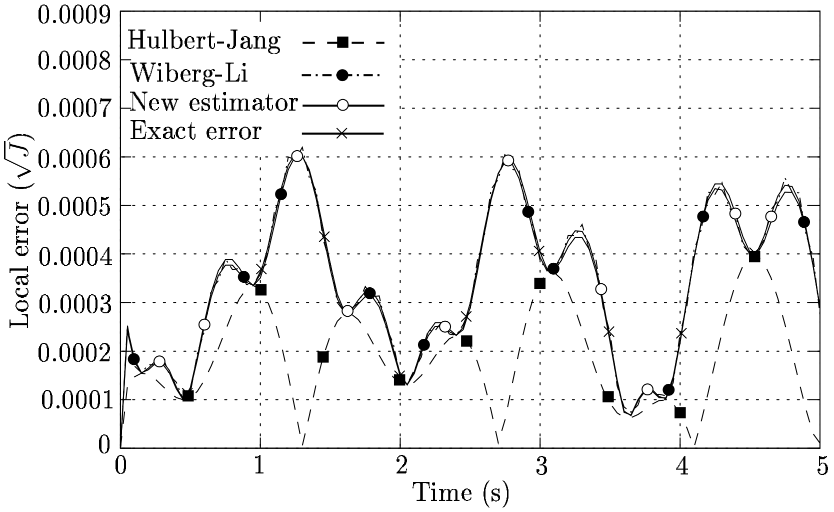


Figure 4.10: One d.o.f. system. Sinusoidal loading. Local error in energy norm with time step $\Delta t = 0.05$ s (trapezoidal rule).

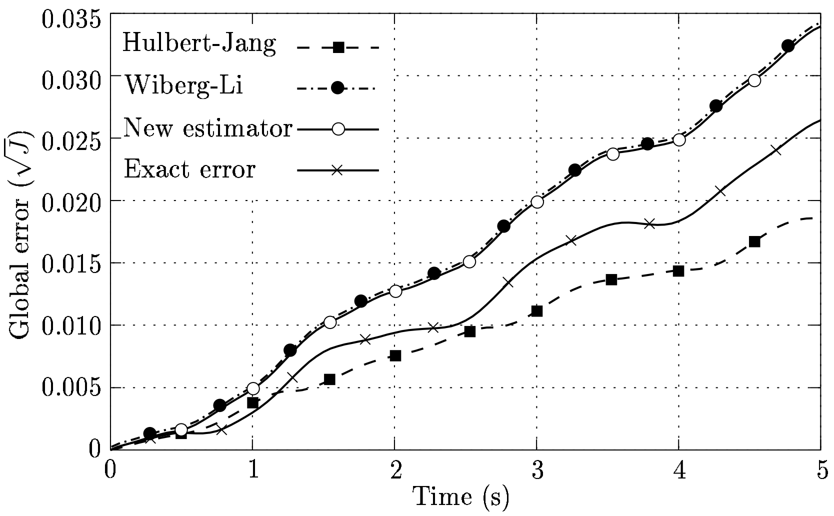


Figure 4.11: One d.o.f. system. Sinusoidal loading. Global error in energy norm with time step $\Delta t = 0.05$ s (trapezoidal rule).

Finally, Figure 4.12 shows the rate of convergence of the error estimation. The three estimators exhibit a second-order rate which is the correct one in a second-order time-stepping method.

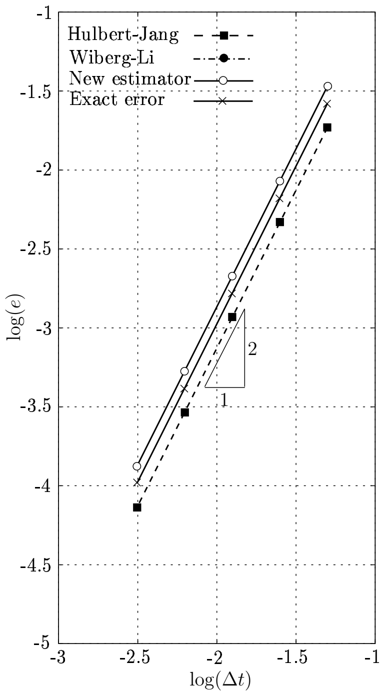


Figure 4.12: One d.o.f. system. Sinusoidal loading. Estimated global errors at time $t = 5.0$ s (trapezoidal rule).

In this second case, we have shown that the method of WIBERG & LI [1993] and the new estimator provide similar results. In contrast, the method of HULBERT & JANG [1995] computes estimates for the error smaller than the exact values.

Triangular load and second-order method

In the third case, we change the loading function. In the two preceding simulations the three estimators have been analyzed in smooth problems. Now, we investigate their performance in a more demanding situation. Specifically, the mass-spring system is excited with a triangular load. The shape of this load is depicted in Figure 4.13. The problem is integrated in time using again the trapezoidal rule and employing a constant time step $\Delta t = 0.05$ s.

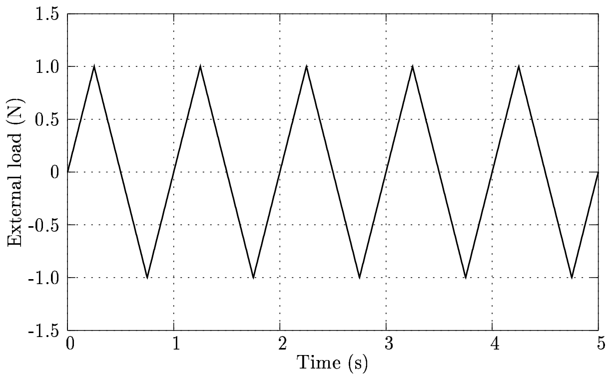


Figure 4.13: Triangular loading function.

We start the analysis of the estimators comparing the local error estimates computed by them. They are depicted in Figure 4.14. The behavior of the estimator of Wiberg & Li [1993] was already analyzed in Section 3.7.2. Recall that it presents sharp jumps in the local estimate at instants where the loading function fails to be differentiable. On the other hand, the estimation provided by the method of Hulbert & Jang [1995] underestimates the true local errors, for the same reasons presented in the previous example (this estimator has a poor performance in second-order time-stepping methods). However, the new estimator continues providing very good local estimates.

These results transfers to the global error, which is shown in Figure 4.15. Whereas the estimator of Hulbert & Jang [1995] does not provide an upper bound for the true error, the new one computes an error very close to the exact value. On the other hand, the error estimated by the method of Wiberg & Li [1993] is one order of magnitude larger than the others. That is, although it provides an upper bound, this is unrealistically large.

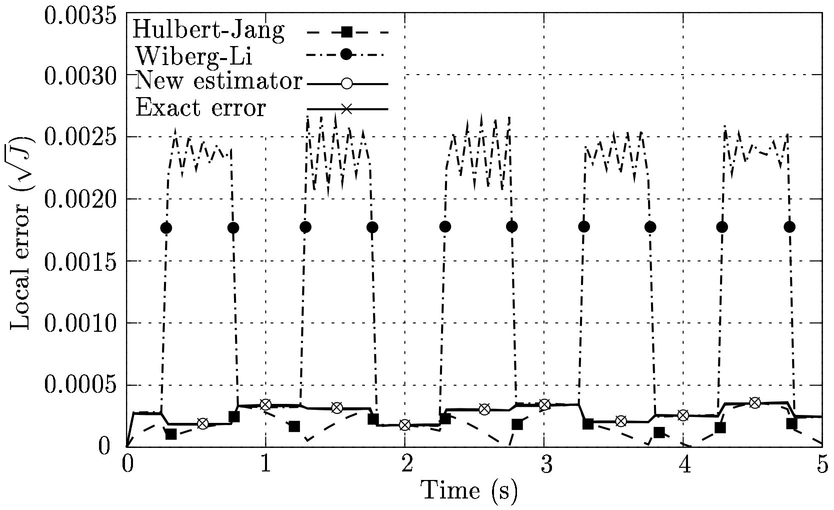


Figure 4.14: One d.o.f. system. Triangular loading. Local error in energy norm with time step $\Delta t = 0.05$ s (trapezoidal rule).

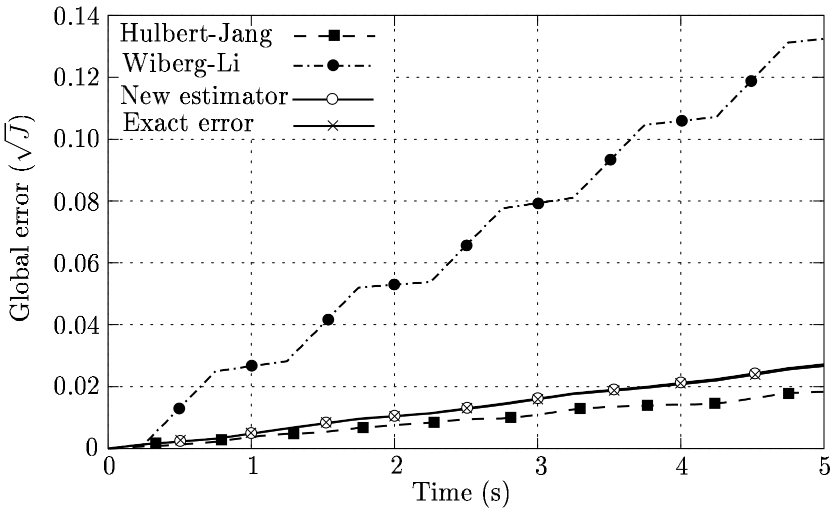


Figure 4.15: One d.o.f. system. Triangular loading. Global error in energy norm with time step $\Delta t = 0.05$ s (trapezoidal rule).

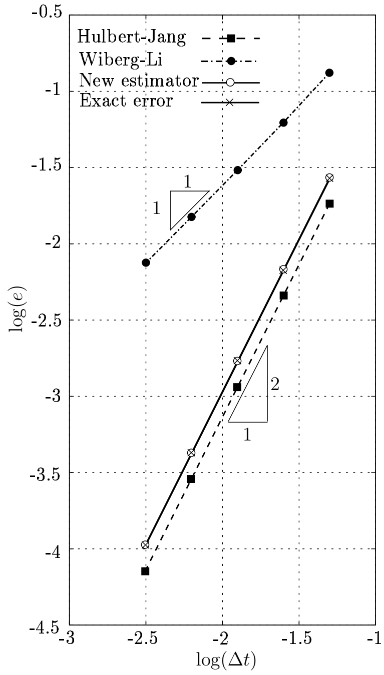


Figure 4.16: One d.o.f. system. Triangular loading. Estimated global errors at time $t = 5.0$ s (trapezoidal rule).

4.5.2 Transient analysis of an elastic deformable block

The previous simulations have been carried out in a simple problem with only one degree of freedom but they have allowed us to extract important conclusions. Henceforth, the simulations will be more complex and as it will be shown, the same conclusions will be reached.

This example is a multi degree of freedom, three dimensional simulation of an elastic deformable block. It has the shape of a truncated pyramid with square top and bottom faces, whose dimensions and finite element mesh are depicted in Figure 4.17. The material of the pyramid is elastic isotropic with Young's modulus $E = 1000$ Pa, Poisson's ratio $\nu = 0.3$, and density $\rho = 1000$ kg/m^3.

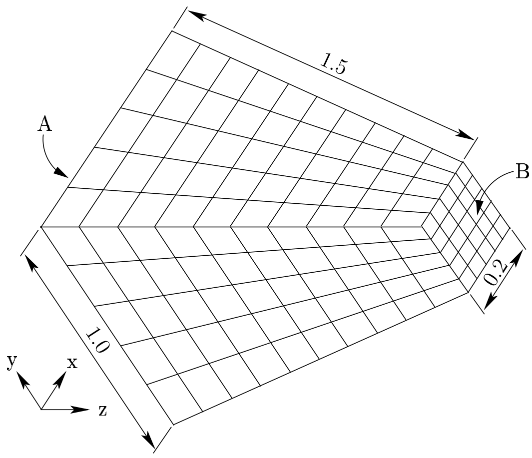


Figure 4.17: Elastic deformable block. Dimensions and finite element mesh.

All the nodes on side A have constrained displacements. There is a normal distributed load on side B whose value is given by a triangular function of peak value 6250 Pa, minimum value 0 Pa, and period 0.4 s. A numerical solution is obtained by integrating in time up to $t = 4$ s with the trapezoidal rule and fixed time step size $\Delta t = 0.0125$ s. A reference solution, assumed to be exact, is obtained by solving the same problem with a much smaller time step $\Delta t = 0.0001$ s.

The results of this example confirm the conclusions obtained in the first numerical example. In Figure 4.18 the local estimated errors are plotted for every time step of the solution. The error estimator of WIBERG & LI [1993] shows again abrupt changes in the estimated errors, coinciding with the instants where the loading function fails to be differentiable. This in turn results in an overestimation of the global error, as can be observed in Figure 4.20, and a wrong convergence rate of the estimated errors (see Figure 4.20). In the same figures we observe that the error estimation furnished by the method of HULBERT & JANG [1995] underestimates the true error although exhibits the correct order. The newly proposed method not only captures the correct order of magnitude of the real errors but provides a guaranteed upper bound of them.

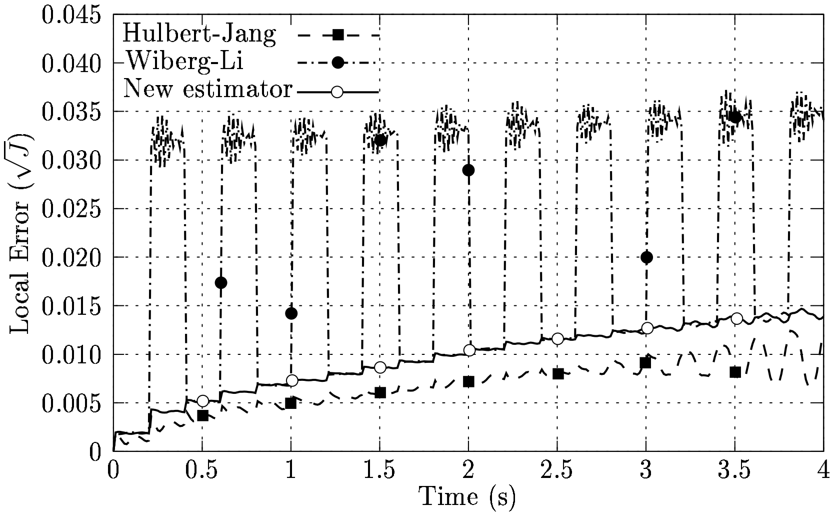


Figure 4.18: Elastic deformable block. Local error in energy norm with time step $\Delta t = 0.0125\,\mathrm{s}$.

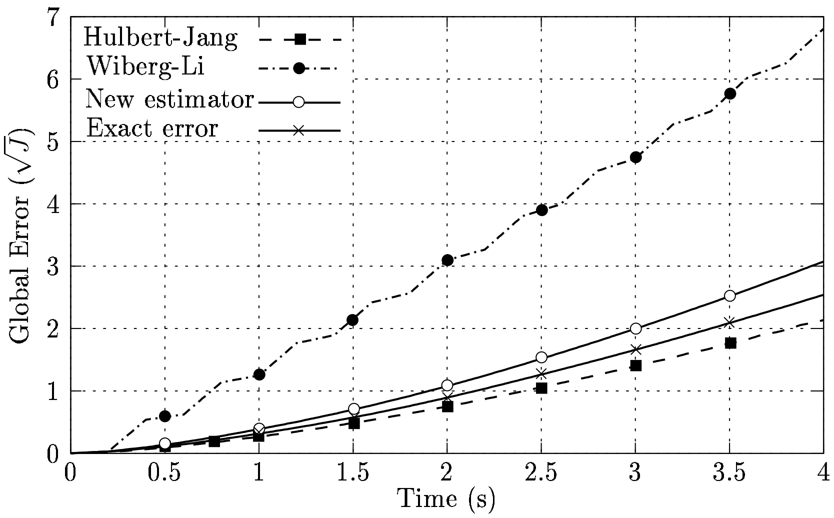


Figure 4.19: Elastic deformable block. Global error in energy norm with time step $\Delta t = 0.0125\,\mathrm{s}$.

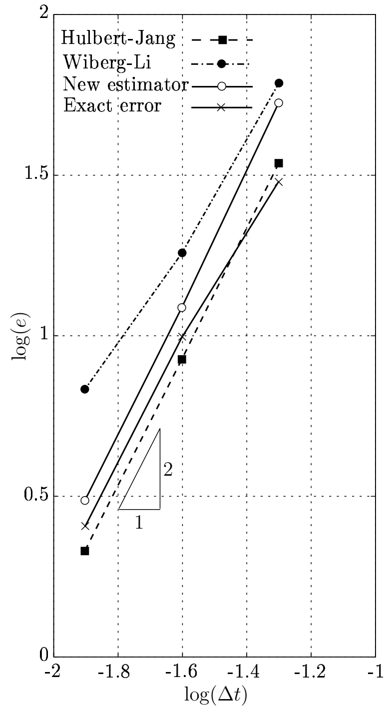


Figure 4.20: Elastic deformable block. Estimated global errors at time $t = 4.0$ s

4.5.3 Simulation of the dynamic response of a train bogie

The third linear simulation is the dynamic analysis of a train bogie subject to an emergency braking. It was first introduced in Lacoma & Romero [2006b]. On the left-hand side of Figure 4.21 is depicted the simplified geometry of the bogie. The shaded areas are the zones where carriage transmits forces to the bogie (zone B) and zones labeled as A are where the bogie joins to the train suspension. The emergency braking is simulated by an application of longitudinal forces whose magnitude is represented on the right-hand side of Figure 4.21. In the finite element model the nodes inside zone A have restricted their displacement.

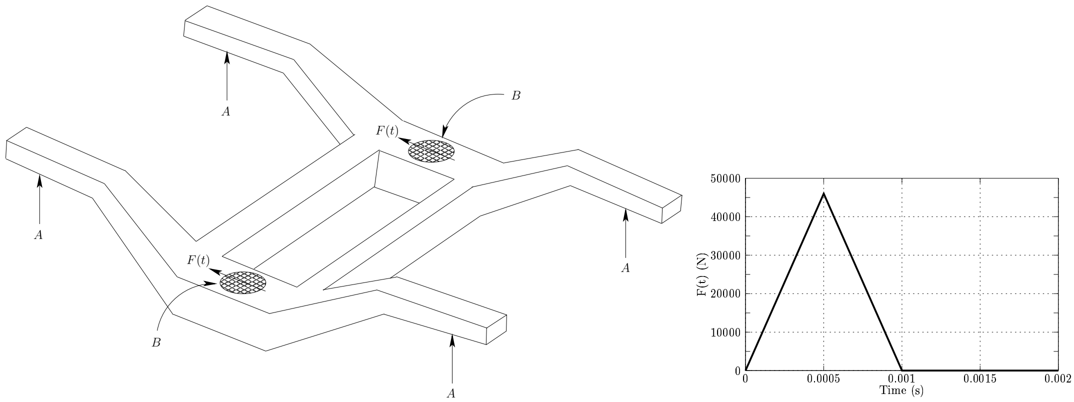


Figure 4.21: Dynamic analysis of a bogie. Left: geometry of the model; Right: Evolution in time of the external load simulating a emergency braking.

In order to perform a dynamical analysis of the bogie, a tridimensional finite element model is built. It is composed of 10780 nodes and 7392 hexaedral elements, each with 8 nodes. The whole model has 31968 degrees of freedom. Figure 4.22 shows the finite element mesh used.

The numerical simulation of the bogie is performed with the trapezoidal rule and the time interval is $[0, 0.002]$ s. A fixed time step $\Delta t = 10^{-6}$ s is considered. In Figure 4.23 several snapshots are depicted showing the deformation which undergoes the bogie.

Figure 4.24 compares the local estimates provided by the three analyzed methods. As previous numerical examples, the estimator of Wiberg & Li [1993] undergoes sharp changes in the local estimation at $t = 0.0005$ s and $t = 0.001$ s, instants where loading function is not derivable. In contrast, both of estimators of Hulbert & Jang [1995] and the new one compute a more smooth local estimate.

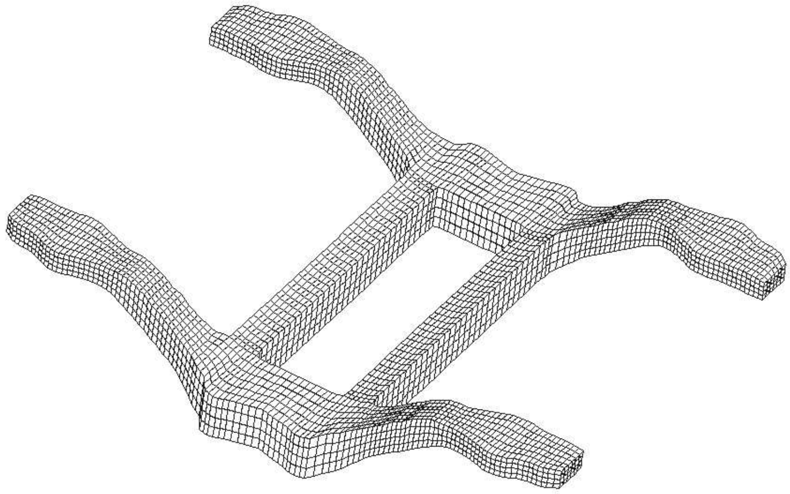


Figure 4.22: Dynamic analysis of a bogie. Finite element mesh of the train bogie.

With respect to the global estimation shown in Figure 4.25, it is clear that the method of WIBERG & LI [1993] has a similar behavior to the others during the first 0.0005 s. However, from this instant, it computes a global error three times larger than computed by the others two estimators. On the other hand, as seen before, we cannot ensure that global estimate given by HULBERT & JANG [1995] is an upper bound, with the risk that it entails.

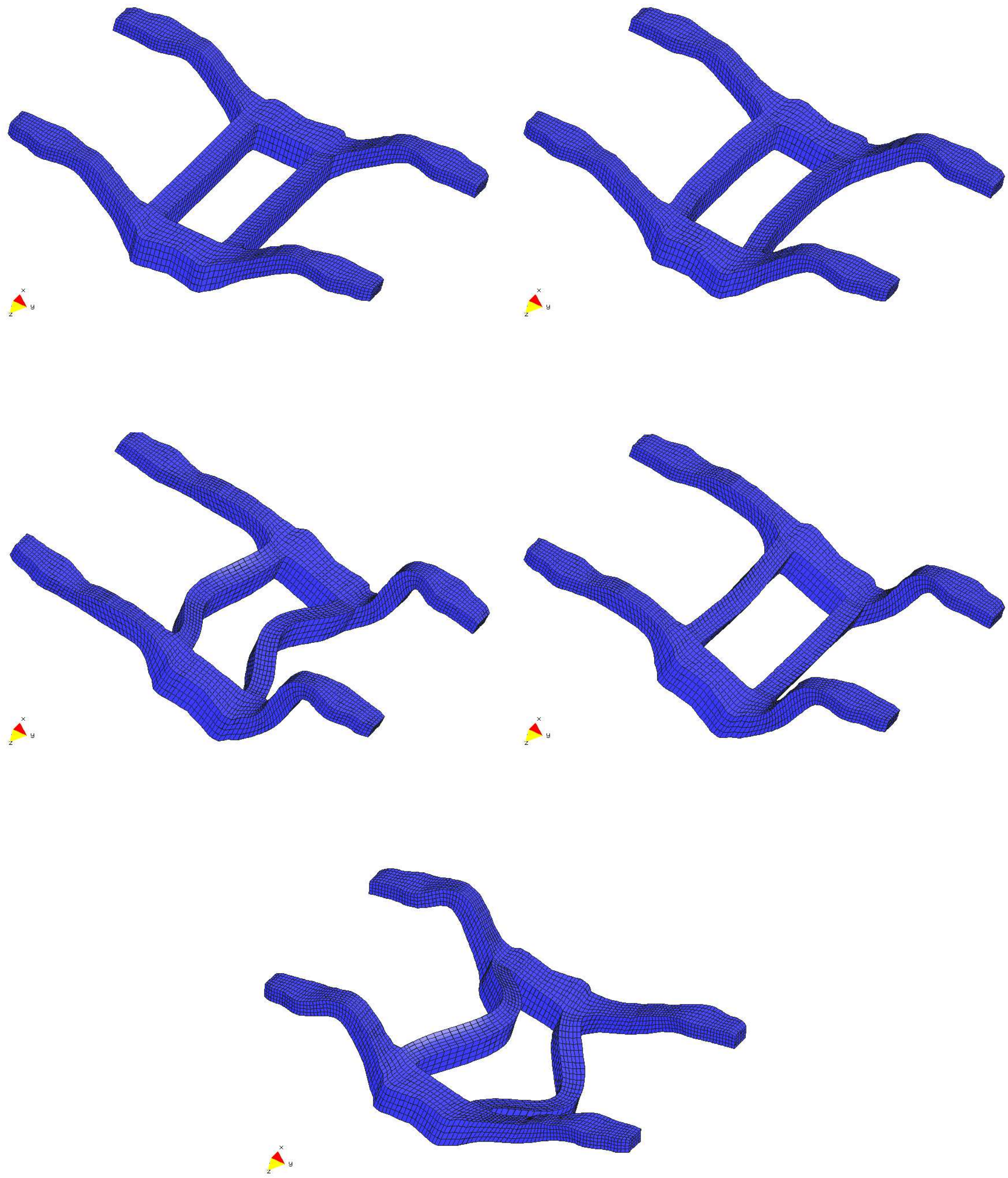


Figure 4.23: Dynamic analysis of a bogie. From to to the bottom, left to right, snapshots of deformed bogie shown every 0.0005 s starting at $t = 0$ s (Deformation $\times 50000$).

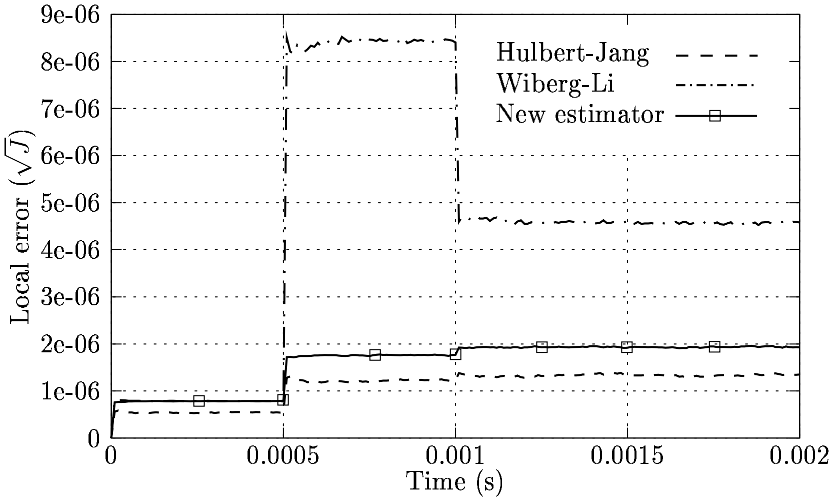


Figure 4.24: Dynamic analysis of a bogie. Estimated local errors in energy norm with time step $\Delta t = 10^{-6}$ s.

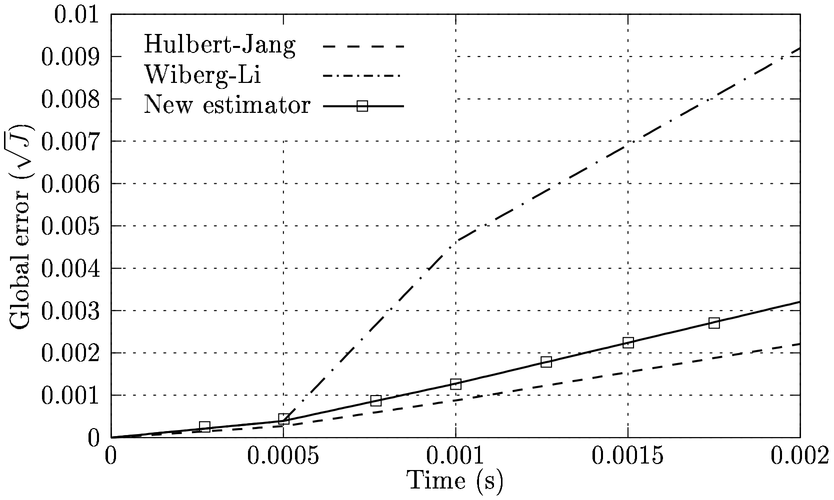


Figure 4.25: Dynamic analysis of a bogie. Estimated global errors in energy norm with time step $\Delta t = 10^{-6}$ s.

4.5.4 Non-linear deformable pendulum

Until now, all numerical simulations presented have been linear analyses. Here, we start the non-linear simulations and we only study the results provided by the new estimator.

The first non-linear simulation is an elastic pendulum, which possesses high and low frequency responses. This example was introduced in Kuhl & Crisfield [1999] to study the performance of several time integration schemes in the non-linear regimen and it is a modification of the simple pendulum problem initially proposed in Bathe [1996] and widely used in different works such as Crisfield [1997], Tarnow [1993] among others.

The pendulum has initial length $l = 3.0443$ m and stiffness $k = 3284.8274$ N/m and it has attached a mass $m = 1$ kg in its extreme. It is dropped from the horizontal position and it moves under the action of gravity. The analysis has been carried out using the HHT method with two different values for parameter α (0.7 and 0.9) and a constant time step $\Delta t = 0.05$ s. The evolution in time of the pendulum is computed up to $t = 8$ s.

In Figure 4.26, the motion of the pendulum is shown where its position is plotted every 0.1 s.

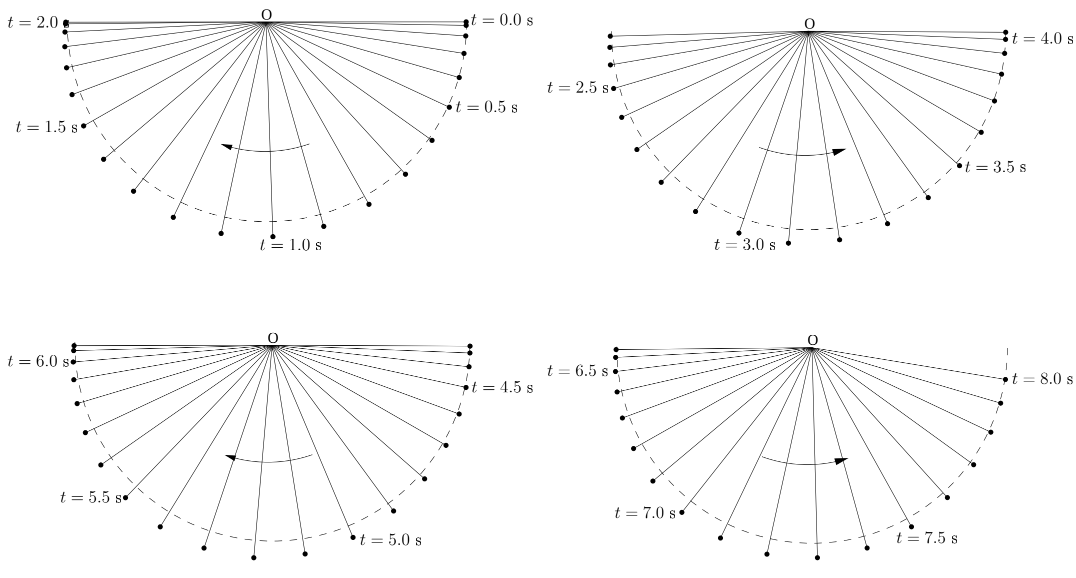


Figure 4.26: Evolution in time of the deformable pendulum. The radial displacement is amplified by a factor of 25. The dashed line would be the trajectory of the pendulum if it were rigid, i.e. if its length remained constant.

Figure 4.27 depicts the evolution of the estimated local error. It is observed that as the pendulum approaches horizontal position, the estimated local error decreases. On the other hand, at instant of highest velocity (when the pendulum is in vertical position) the local error reaches the maximum value.

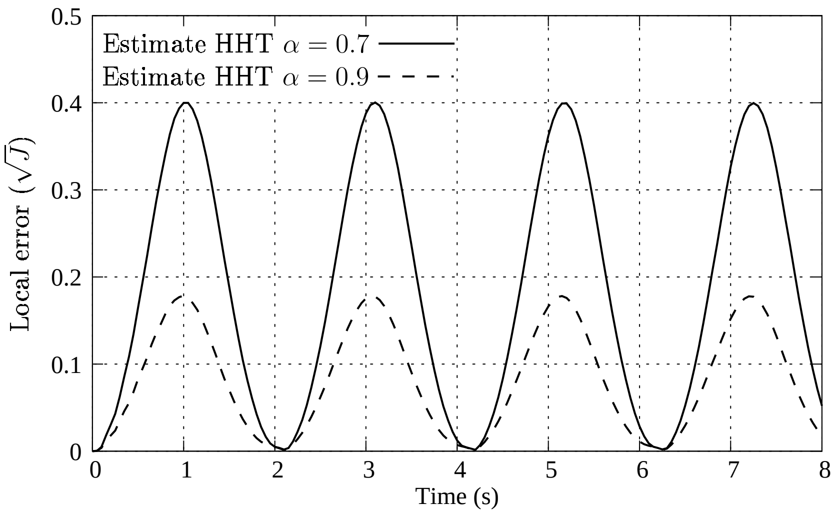


Figure 4.27: Non-linear deformable pendulum. Estimated local errors in energy norm with time step $\Delta t = 0.05$ s.

In Figure 4.28 the estimated global errors are plotted and they are compared to the "exact" ones. It can be observed that, in both cases, the estimator provides an accurate and safe upper bound for the "exact" error. The "exact" solution has been approximated solving the problem with a much smaller time step $\Delta t = 0.0001$ s.

Finally, Figure 4.29 shows the rate of convergence of both the estimated global error and the "exact" error measured in the energy norm. It is known the HHT method is second-order accurate. Hence, the global error predicted by an estimator must exhibit this same rate of convergence. From Figure 4.29 this fact can be verified.

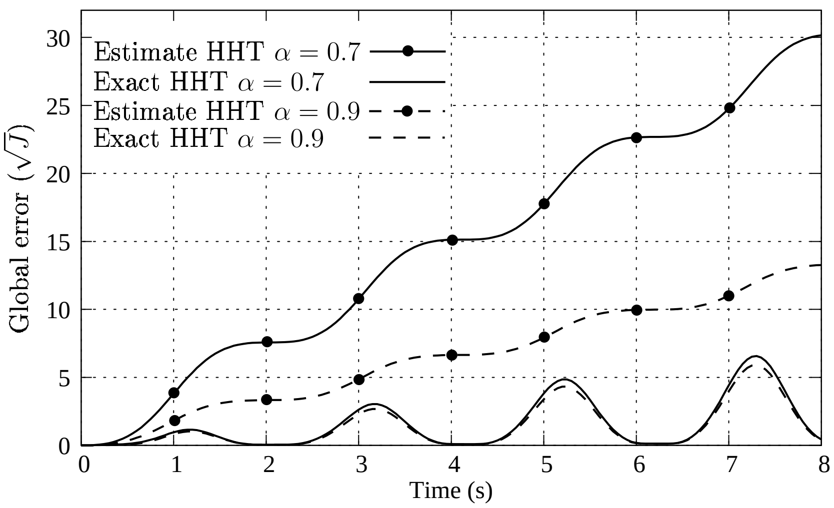


Figure 4.28: Non-linear deformable pendulum. Global errors in energy norm with time step $\Delta t = 0.05$ s.

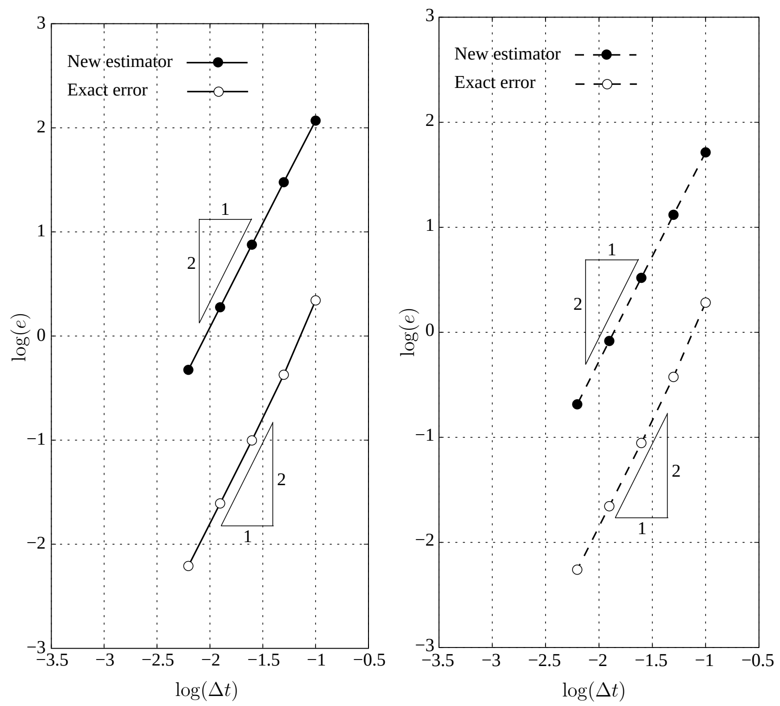


Figure 4.29: Non-linear deformable pendulum. Estimated and exact global errors at time $t = 8.0$ s. (Left: HHT method with $\alpha = 0.7$. Right: HHT method with $\alpha = 0.9$)

4.5.5 Dynamic behavior of a car suspension

The second non-linear example is a more realistic three dimensional multi-body system with several degrees of freedom. This mechanism simulates a car suspension and it is composed of five deformable bars, a coil spring and a damper. More precisely, four bars form two parallel isosceles triangles (AEB and CFD) joined by the vertical bar EF. In Figure 4.30, the exact geometry of the model and position of each point is shown.

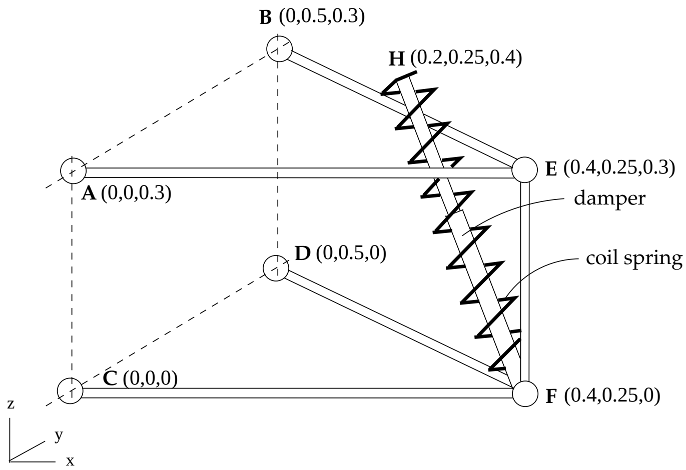


Figure 4.30: Geometry of a car suspension (positions of points are expressed in meters).

The bars undergo large displacements and, thus, the resulting problem is non-linear. The points A, B, C, D and H are fixed, but they allow rotation around them. Between the points H and F there is a coil spring and a damper, with stiffness $k = 10^4$ N/m and damping $c = 10^3$ Ns/m, respectively. All bars are supposed to be deformable with the following mechanical properties: axial stiffness per unit length $EA = 2.1 \cdot 10^{10}$ N/m, Poisson's coefficient $\nu = 0.3$ and density $\rho = 7890$ kg/m^3.

At the point F, a vertical external load is applied in the vertical direction (z-direction). Initially, the car suspension is charged up to 500 N. Then, from time $t = 1$ s to $t = 2$ s three consecutive peaks of 1000 N are introduced. Evolution in time of this external load is plotted in Figure 4.31.

Several transient analysis will be carried out by means of the HHT method with three different values for parameter α (0.95, 0.90 and 0.80), and a constant time step $\Delta t = 0.01$ s.

Figure 4.32 depicts the evolution of local errors computed using the estimator

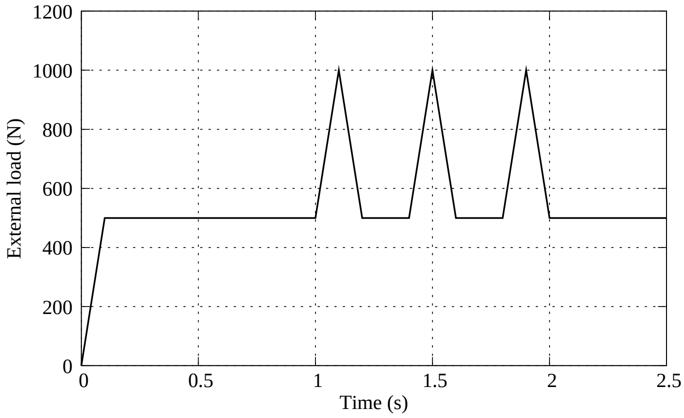


Figure 4.31: Evolution in time of external load applied to the car suspension.

detailed in Section 4.4. It can be observed that smaller the value of parameter α is, larger the local error obtained by the estimator is.

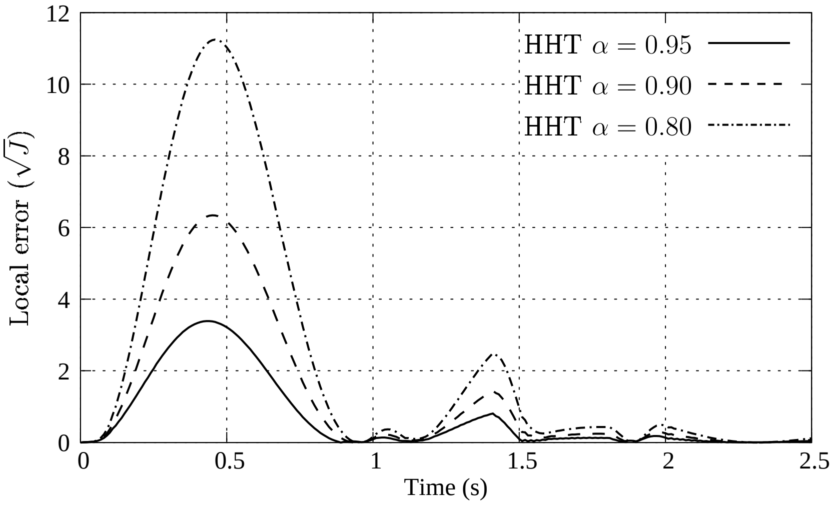


Figure 4.32: Car suspension. Local error with time step size $\Delta t = 0.01$ s.

Next, estimates for the global error are compared with “exact” ones obtained solving the problem with a much smaller time step $\Delta t = 0.00001$ s. On the top of Figure 4.33, the evolution of global error estimates is plotted. Just like in the case of local errors, simulation performed with the HHT method with α equal to 0.95 provides more accurate results. Global error estimations for the three different simulations are upper bounds of exact errors, which are presented on the bottom of Figure 4.33.

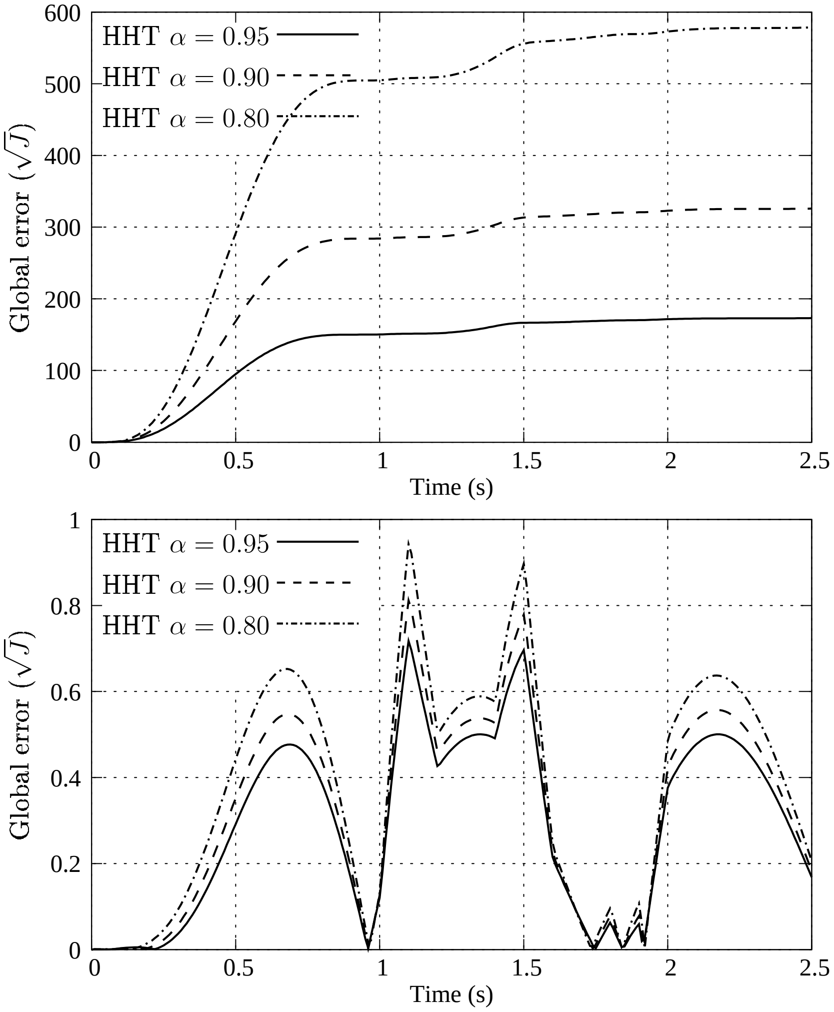


Figure 4.33: Car suspension. Global error with time step size $\Delta t = 0.01$ s. (Top: estimated errors. Bottom: exact errors).

In Figure 4.34 it is shown the order of convergence of global errors. All of them overestimate the exact error being, thus, safe bounds. Also, they present a correct second order of convergence

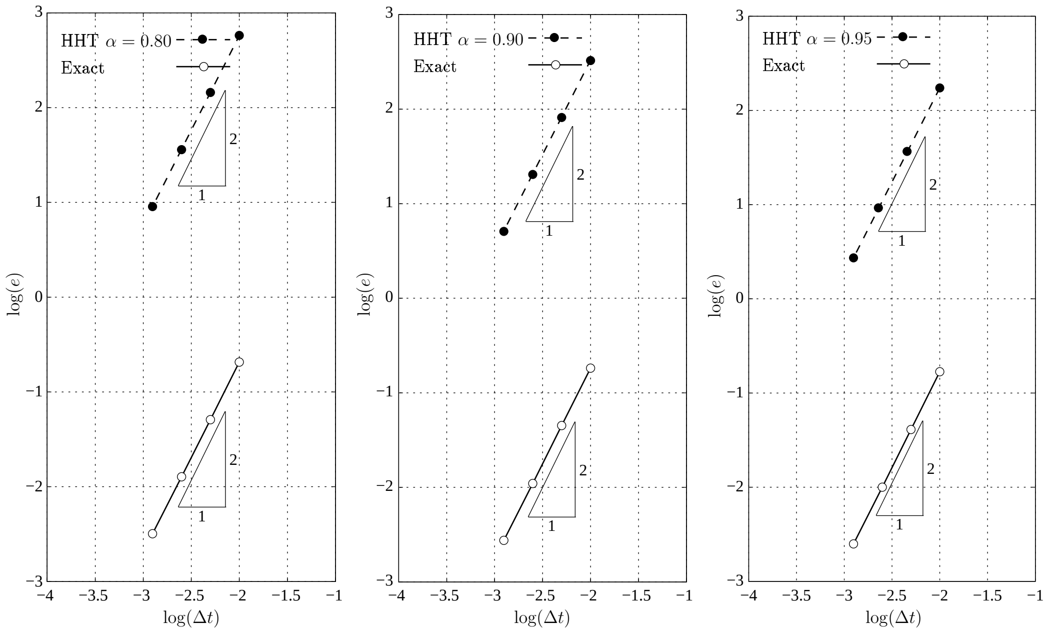


Figure 4.34: Car suspension. Global errors at instant $t = 2.5$ s. (Left HHT with $\alpha = 0.80$. Middle: HHT with $\alpha = 0.90$. Right: HHT with $\alpha = 0.95$).

4.5.6 Three deformable balls bouncing inside a rigid box

This third non-linear and last simulation is the most complex one. It consist of three deformable and hollow balls moving and bouncing inside a rigid box.

The three balls are identical with radius of 1 m, thickness 0.1 m and they are subjected to an internal pressure $p = 10^4$ Pa. The material of the balls is elastic with Young's modulus $E = 6 \cdot 10^4$ Pa, Poisson's ratio $\nu = 0.3$ and density $\rho = 2000$ kg/m^3. They are initially placed at coordinates $(0, 0, 0)$ m the red ball, $(0, 3, 0)$ m the green ball and $(0, -3, 0)$ m the blue one, and they are released with initial velocities $(0.1, 0, -0.17321)$ m/s, $(0.14142, 0, 0.14142)$ m/s, $(0.11547, 0.11547, -0.11547)$ m/s, respectively (all expressed in Cartesian coordinates). The finite element model of each individual ball has 600 quadrilateral elements with a standard membrane formulation and 602 nodes for a total of 5418 degrees of freedom.

The rigid box is a rectangular parallelepiped with dimensions $9 \times 9 \times 6$ m^3 and its center is at coordinates $(0, 0, 0)$. Balls collide between them and also against the walls of the box. All contacts are imposed by a penalty method.

The transient simulation is carried out using the HHT method with parameter $\alpha = 0.7$ (maximum numerical damping) and a constant time step $\Delta t = 0.025$ s. The motion is simulated during 100 s.

In Figures 4.35 and 4.36 it is plotted the position of three balls every 10 s. As can be seen, the balls undergo a large deformations when they collide against the rigid walls (see, e.g., in Figure 4.35 the green ball at time $t = 30$ s).

The goal of this simulation is to show that the new error estimator developed for the HHT method can be used in complex non-linear problems, such as, the ones with large displacements and contacts. Figure 4.37 depicts the evolution of the estimated local error. When a contact between two balls or against a wall take place, it is produced a peak in the error estimation. However, during a period without contacts,e.g., in the time interval $(70, 90)$ s, the error estimator provides a smoother response.

Adding up the estimated local errors, a bound for the global error can be computed. It is plotted in Figure 4.38. To study its accuracy we compare it to an "exact" error. This "exact" error is obtained subtracting the solution provided by the HHT method with a reference solution calculated with a smaller time step $\Delta t = 0.0005$ s. Then, the energy norm of this "exact" error is computed. Note the good performance of the estimator since it provides an safety and accurate upper bound for the exact error.

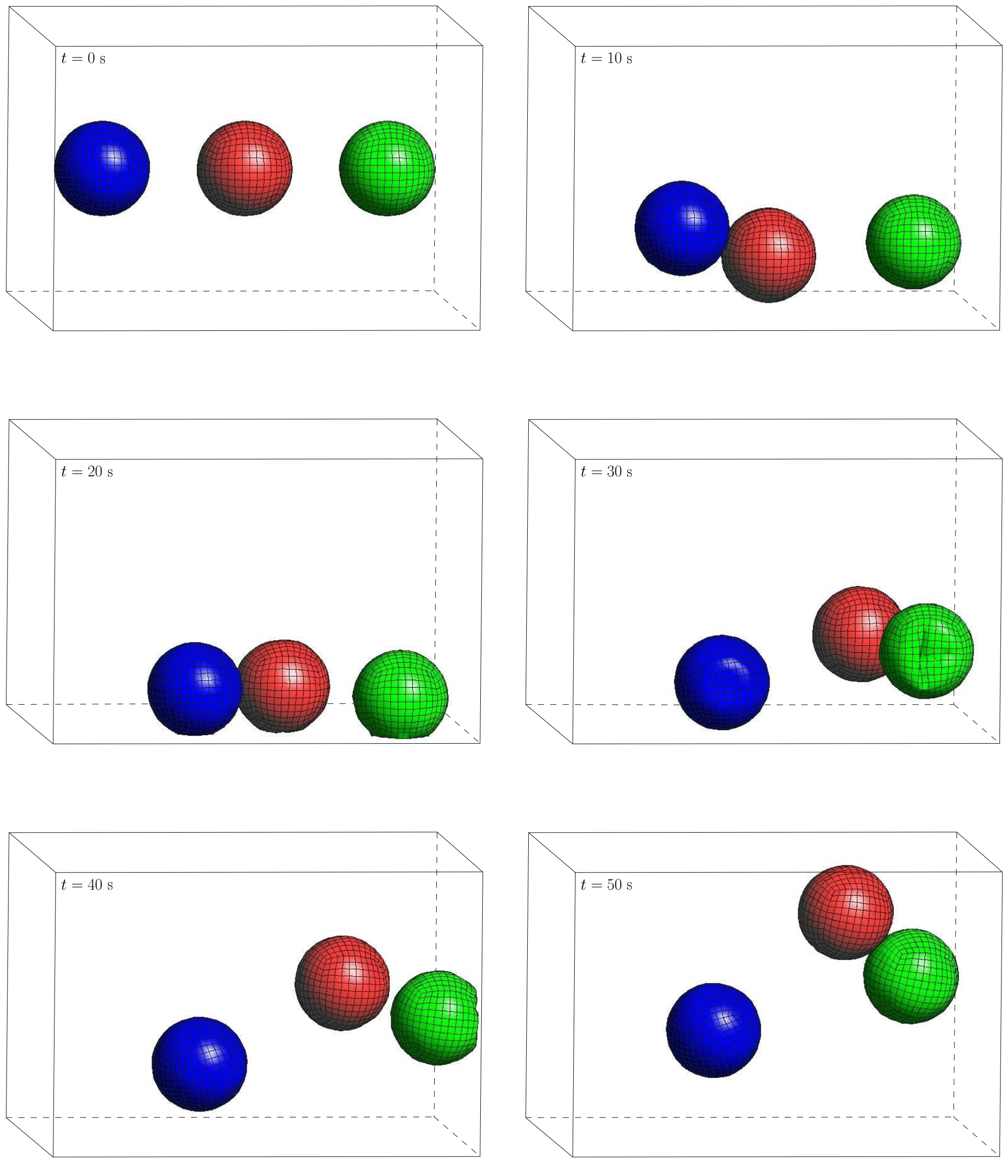


Figure 4.35: Deformable bouncing balls. Solution obtained with the HHT method ($\alpha = 0.7$) and a constant time step $\Delta t = 0.025$ s. Snapshots shown every 10 seconds starting at $t = 0$ s.

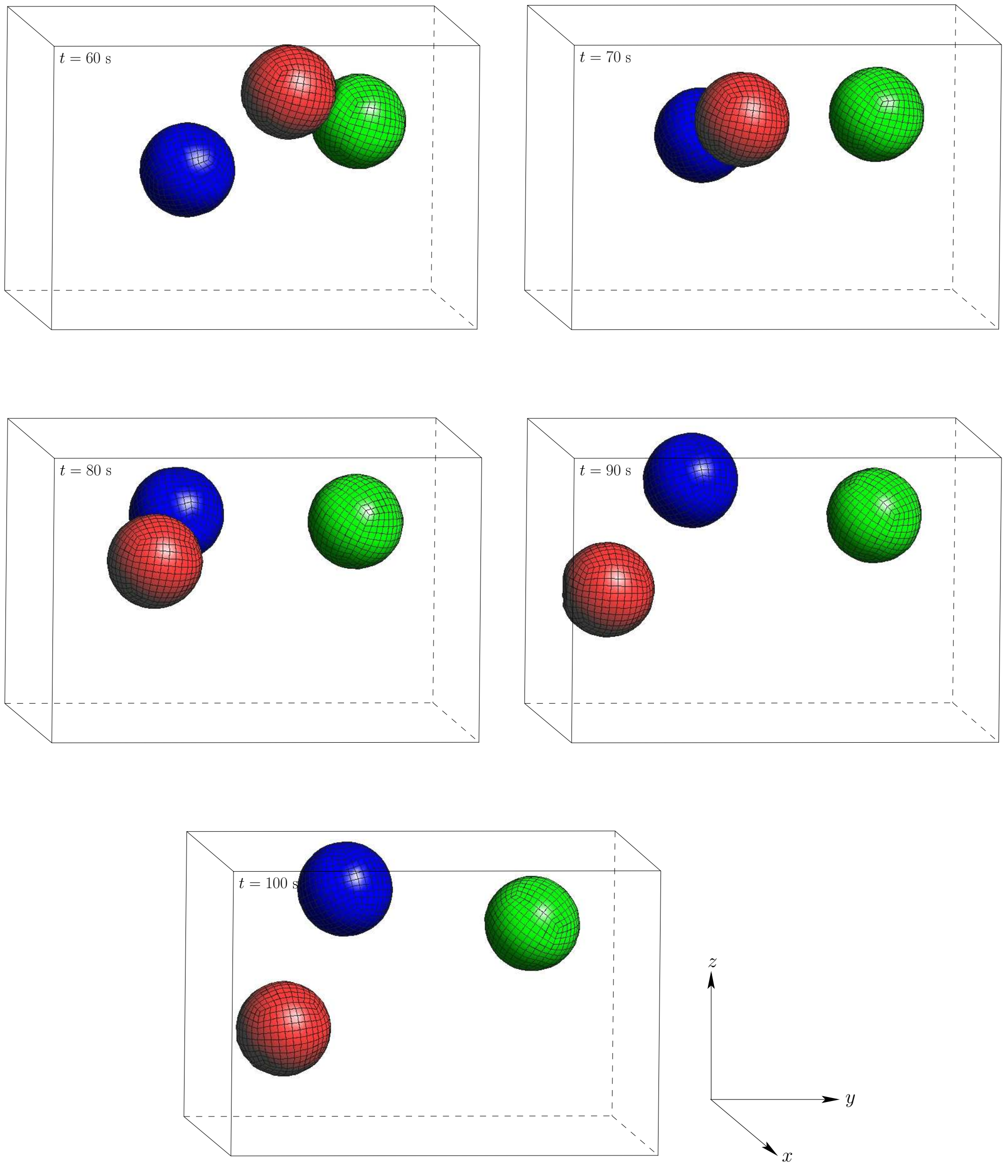


Figure 4.36: Deformable bouncing balls . Solution obtained with the HHT method ($\alpha = 0.7$) and a constant time step $\Delta t = 0.025$ s. Snapshots shown every 10 seconds starting at $t = 60$ s.

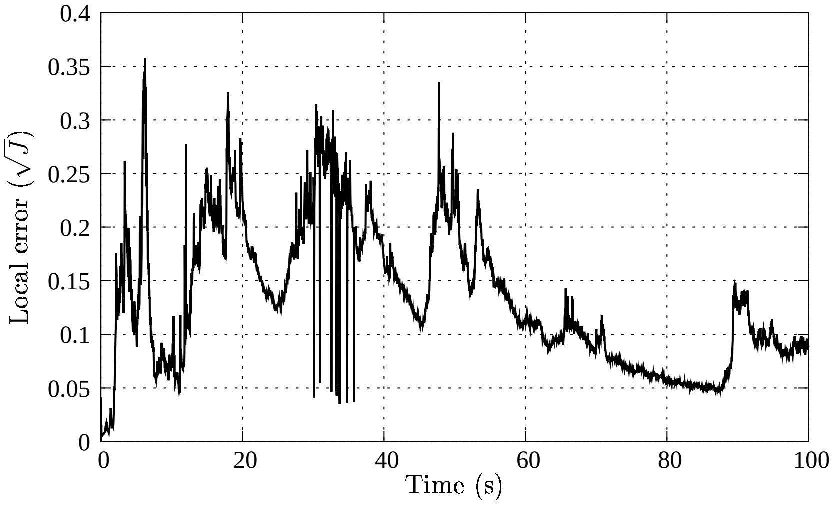


Figure 4.37: Deformable bouncing balls. Estimated local error in energy norm with time step $\Delta t = 0.025$ s.

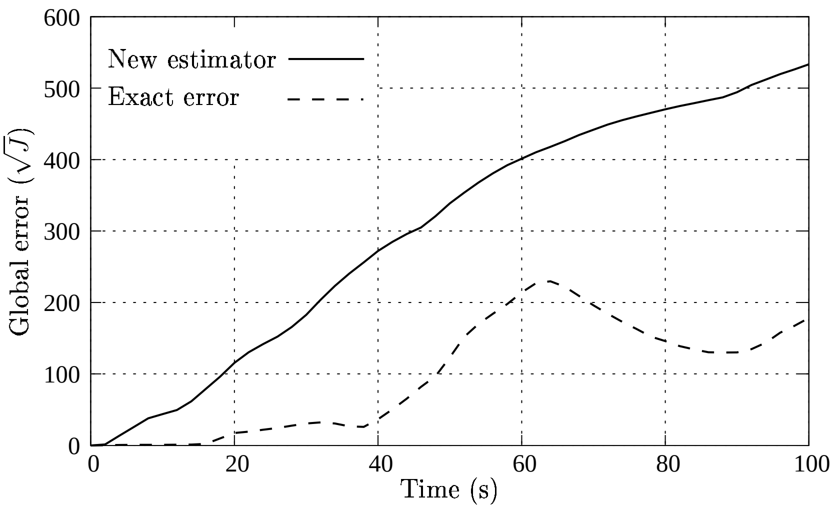


Figure 4.38: Deformable bouncing balls. Global errors in energy norm with time step $\Delta t = 0.025$ s.

Appendix 4.I

In this appendix, we study the order of approximation of the acceleration provided by the HHT method to the local exact value at t_{n+1}.

Proposition 4.3. *The acceleration computed by the HHT method is a first order approximation to the exact acceleration of the local problem at t_{n+1}, i.e.,*

$$\mathbf{A}_{n+1} = \ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) + \mathcal{O}(\Delta t) \ . \tag{4.48}$$

PROOF: This proof follows the same steps as in Proposition 4.1. Using the balance equation and expanding $\boldsymbol{u}^{[n]}(t_{n+1})$ and $\dot{\boldsymbol{u}}^{[n]}(t_{n+1})$ by Taylor series about t_n and using the initial conditions of the local problem defined in (3.2), one obtains

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] = {} & \mathbf{F}_{n+\alpha} - (1-\alpha)\left[\mathbf{C}\mathbf{V}_n + \mathbf{K}\mathbf{U}_n\right] - \mathbf{M}\ddot{\boldsymbol{u}}^{[n]}(t_{n+1}) \\ & - \alpha\mathbf{C}\left[\mathbf{V}_n + \Delta t\ddot{\boldsymbol{u}}^{[n]}(t_n) + \frac{\Delta t^2}{2}\dddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] \\ & - \alpha\mathbf{K}\left[\mathbf{U}_n + \Delta t\mathbf{V}_n + \frac{\Delta t^2}{2}\ddot{\boldsymbol{u}}^{[n]}(t_{n+1})\right] + \mathcal{O}(\Delta t^3) \ .\end{aligned} \tag{4.49}$$

Simplifying terms and expanding the acceleration $\ddot{\boldsymbol{u}}(t_{n+1})$ by Taylor series about t_n up to order one, the following expression is reached,

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}(t_{n+1})\right] = {} & \mathbf{F}_{n+\alpha} - \mathbf{C}\mathbf{V}_n - \mathbf{K}\mathbf{U}_n - \alpha\Delta t\mathbf{C}\ddot{\boldsymbol{u}}(t_n) - \alpha\Delta t\mathbf{K}\mathbf{V}_n \\ & - \mathbf{M}\left[\ddot{\boldsymbol{u}}(t_n) + \Delta t\dddot{\boldsymbol{u}}(t_n)\right] + \mathcal{O}(\Delta t^2) \ .\end{aligned} \tag{4.50}$$

Using the balance equation $(3.2)_1$ at time t_n, an reordering terms yields

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1} - \ddot{\boldsymbol{u}}(t_{n+1})\right] = {} & \mathbf{F}_{n+\alpha} - \mathbf{F}(t_n^+) - \alpha\Delta t\left[\mathbf{C}\ddot{\boldsymbol{u}}(t_n) + \mathbf{K}\mathbf{V}_n\right] \\ & - \Delta t\mathbf{M}\dddot{\boldsymbol{u}}(t_n) + \mathcal{O}(\Delta t^2) \ .\end{aligned} \tag{4.51}$$

Finally, expanding $\mathbf{F}_{n+\alpha}$ in terms of its convex combination and taking into account the time derivative of the balance equation,it is concluded that

$$\begin{aligned}\mathbf{M}\left[\mathbf{A}_{n+1}-\ddot{\boldsymbol{u}}(t_{n+1})\right] &= \alpha\left[\mathbf{F}(t_{n+1}^{-})-\mathbf{F}(t_{n}^{+})\right] \\ &\quad -\alpha\Delta t\dot{\mathbf{F}}(t_{n}^{+})-\Delta t(1-\alpha)\mathbf{M}\dddot{\boldsymbol{u}}(t_n)+\mathcal{O}(\Delta t^2) \\ &= -\Delta t(1-\alpha)\mathbf{M}\dddot{\boldsymbol{u}}(t_n) \\ &\quad +\alpha\underbrace{\left[\mathbf{F}(t_{n+1}^{-})-\mathbf{F}(t_{n}^{+})-\Delta t\dot{\mathbf{F}}(t_{n}^{+})\right]}_{\mathcal{O}(\Delta t^2)}+\mathcal{O}(\Delta t^2)\ .\end{aligned} \tag{4.52}$$

Thus,

$$\mathbf{A}_{n+1}-\ddot{\boldsymbol{u}}(t_{n+1}) = -\Delta t(1-\alpha)\dddot{\boldsymbol{u}}(t_n)+\mathcal{O}(\Delta t^2)=\mathcal{O}(\Delta t) \tag{4.53}$$

■

Chapter 5

Adaptive methods in time for solid dynamic problems

Time-step selection algorithms from the point of view of automatic control theory

In the previous chapters we have presented some algorithms for estimating the error in problems of deformable solid dynamics. Now, they will be used as starting point for the development of adaptive schemes.

An adaptive scheme is, within the context of a transient simulation, an algorithm that selects automatically the time step size to fulfill two requirements. The first one is to keep the error below a tolerance. That is, the analysis must be solved with the desired accuracy. The second requirement is the efficiency. In principle, the larger the time step size is, the fewer steps will be needed to calculate and therefore, the computational cost will be reduced.

In Section 5.1, we start addressing a standard adaptive algorithm which is widely employed. Section 5.2 states the problem of selecting the time step size from the point of view of control theory. This approach transforms the issue of choosing the time step size into a issue of determining what type of controller is better. The advantage is that all techniques used in control theory can be applied for our purpose. In Section 5.3 different types of controllers are discussed and in Section 5.4 optimum values for the parameters which define these controllers are given. These values are obtained from an analysis of the system in the frequency domain. Section 5.6 gives in detail an implementation for the proposed adaptive strategies. In Section 5.7, we discuss the performance of the different adaptive strategies through three numerical examples. Specifically, Section 5.7.1 describes the results obtained for a cantilever beam subject to a triangular loading. Section 5.7.2 considers the bounces of a deformable ball against a rigid plane. Finally, in Section 5.7.3 a more real simulation is performed. It is the analysis of the impact of a dummy against an airbag.

5.1 A simple standard formula for selecting the time step size

Many of the existing works in the literature on the selection of the time step size in general purpose ODE solvers or in transient solid dynamic simulations adopt, with slightly modifications, the same strategy. See among others the classical book by Gear [1971] and the articles by Zienkiewicz & Xie [1991], Zeng et al. [1992], Choi & Chung [1996], Li & Wiberg [1998], Chung et al. [2003] and Hulbert & Jang [1995]. The aim of this section is to introduce briefly this strategy.

As explained in Chapter 2 (Section 2.3.1), the local error for a p-order accurate time-stepping method is of size $\mathcal{O}(\Delta t^{p+1})$. Therefore, the estimated local error at t_{n+1} can be written as

$$\Theta_{n+1} = c \cdot \Delta t_n^{p+1}, \qquad \text{with}\, c \in \mathbb{R}^+ \ . \tag{5.1}$$

Suppose now that we have solved a dynamic analysis up to time t_n and we want to advance one step more up to t_{n+1}. Then, the new time step Δt_n can be selected using the next formula:

$$\Delta t_n = \left(\frac{\varepsilon}{\Theta_n} \right)^{\frac{1}{p+1}} \cdot \Delta t_{n-1} \ , \tag{5.2}$$

where ε is a user-defined tolerance, Θ_n is the error estimate at t_n and p is the order of accuracy of the time integration method.

Remark 5.1. Justification of formula given in (5.2):

If ε is the desired error (or tolerance) at t_{n+1}, it can be expressed as

$$\varepsilon = c \cdot \Delta t_n^{p+1} \ . \tag{5.3}$$

Dividing the expression (5.1) by (5.3), the formula (5.2) is immediately obtained.
□

Formula (5.2) is used to, once a step is accepted, select the next time step. Otherwise, that is, when the estimated error exceeds the tolerance, the step is rejected and computed again with a smaller size. In such situation, Formula (5.2) can be again employed, but replacing Δt_n and Δt_{n-1} by Δt_n^{new} and Δt_n^{old}, respectively, resulting

$$\Delta t_n^{\text{new}} = \left(\frac{\varepsilon}{\Theta_n} \right)^{\frac{1}{p+1}} \cdot \Delta t_n^{\text{old}} \ , \tag{5.4}$$

This simple strategy works well in smooth problems. However, in non-smooth problems it follows an oscillating or "nervous" behaviour which is shown, e.g., in GUSTAFSSON [1991]. This "nervous" behaviour leads to excessively large increments of the steps, causing that many steps must be rejected and damaging thus, the efficiency of the adaptive scheme.

As pointed in GUSTAFSSON [1991] and SÖDERLIND [2002], this strategy can be viewed as a controller of type integral within control theory. This aspect is investigated throughout this chapter.

5.2 Modeling of the process of selecting the time step size

In this section, we build a mathematical model for the selection of the time step size in time-stepping methods based on control theory.

First, we start recalling the initial value local problem which was formulated in (3.3):

$$\begin{cases} \dot{\boldsymbol{y}}^{[n]}(t) = \mathbf{B}\boldsymbol{y}^{[n]}(t) + \mathbf{g}(t) \qquad t \in \mathcal{I}_n = [t_n, t_{n+1}] \ , \\ \quad \boldsymbol{y}^{[n]}(t_n) = \tilde{\boldsymbol{z}}_n \ , \end{cases} \tag{5.5}$$

remembering that the objective is to solve this problem with the desired accuracy.

It can be shown that the exact local error at t_{n+1} can be expressed as (see, e.g. GEAR [1971]):

$$\boldsymbol{e}_{n+1} = \boldsymbol{\Phi}(t_n, \tilde{\boldsymbol{z}}_n) \cdot \Delta t_n^{p+1} + \mathcal{O}(\Delta t^{p+2}) \ , \tag{5.6}$$

where $\mathbf{\Phi}(t_n, \tilde{\boldsymbol{z}}_n)$ is the so-called *principal error function* and p is the order of accuracy of the numerical method employed to carry out the simulation.

Following the same scheme as in (5.6), a similar expression for the estimated local error at t_{n+1} can be written,

$$\boldsymbol{\theta}_{n+1} = \mathbf{\Phi}_n \cdot \Delta t_n^{p+1} + \mathcal{O}(\Delta t^{p+2}) \ . \tag{5.7}$$

By taking the energy norm in both sides of (5.7), it follows

$$\Theta_{n+1} := ||\boldsymbol{\theta}_{n+1}||_E \approx \phi_n \Delta t_n^k \ , \tag{5.8}$$

where it is taken into account that the second term in (5.7) is negligible compared to the first one. Also for simplicity, we have introduced the notation: $\phi_n := ||\mathbf{\Phi}_n||_E$ and $k := p + 1$.

Remark 5.2. Expression given by (5.6) holds in the asymptotic regime, i.e., when the time step size tends to zero. In a numerical simulation, with a finite value for the time step this model cannot maybe fulfil. It will be discussed with more details in Section 5.5. □

As we are going to focus on linear control theory, expression (5.8) is transformed, by taking logarithms, to a linear relation. Thus,

$$\Theta_{n+1} \approx \phi_n \Delta t_n^k \Longrightarrow \log \Theta_{n+1} = \log \phi_n + k \log \Delta t_n \ , \tag{5.9}$$

where the sign of approximation is substituted by a sign of equality to make easier the notation. Indeed, if the terms $\log \Theta_{n+1}$, $\log \phi_n$ and $\log \Delta t_n$ are taken as new variables then, relation (5.9) is linear.

Relation (5.9) models the system which we want to control and it is referred to as *plant*. In control theory it is very common to represent the different mathematical expressions under study in a graphical way by means of block diagrams. This representation allows to observe in a simple manner the relations and connections between all parts of the system and also, a better comprehension of them. Following this procedure, equation (5.9) is represented as follows.

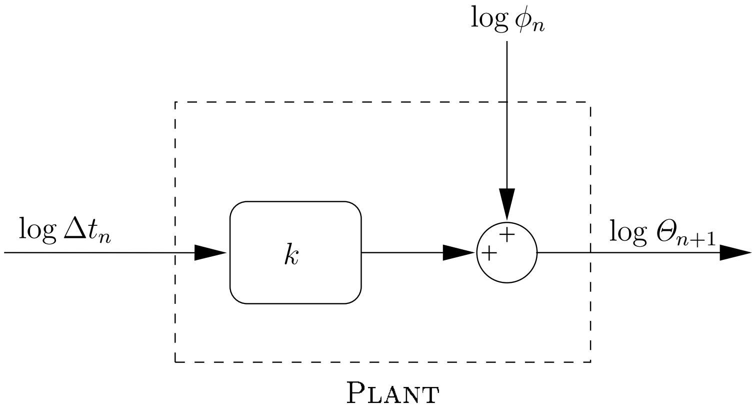


Figure 5.1: Block diagram of the open loop system of selection of time step size.

In this figure, we can observe the different parts which compose the system. These are the *plant*, the *inputs* ($\log \Delta t_n$ and $\log \phi_n$) and the *output* ($\log \Theta_{n+1}$). This type of system is referred to as *open-loop* system in contrast to *closed-loop* systems which will be introduced later. The term *open-loop* means that there does not exist any feedback between different parts of the system.

The open loop system can be seen as a black box which computes an estimation for the local error in function of the time step size.

In mathematical terminology, equation (5.9) is known as a *difference equation*. It is for discrete systems as differential equations for continuous systems. To handle in an easier way the difference equation (5.9), it is transformed to an algebraic equation via the the Z-transform* (see, e.g. Phillips & Nagle [1998] for more details about the Z-transform).

Denoting by $\mathcal{Z}[\vartheta_n](z)$ the Z-transform of sequence ϑ_n and transforming to the z-domain the equation (5.9), it follows that

$$\begin{aligned}
&\mathcal{Z}[\log \Theta_{n+1}](z) = \mathcal{Z}[\log \phi_n + k \log \Delta t_n](z) \\
\Rightarrow &z\mathcal{Z}[\log \Theta_n](z) = \mathcal{Z}[\log \phi_n](z) + k\mathcal{Z}[\log \Delta t_n](z) \\
\Rightarrow &z \log \Theta_z = \log \phi_z + k \log \Delta t_z \\
\Rightarrow &\log \Theta_z = z^{-1} \log \phi_z + G_p(z) \log \Delta t_z,
\end{aligned} \tag{5.10}$$

where $G_p(z) = kz^{-1}$ is the transfer function of the plant and $\log \Theta_z$, $\log \phi_z$ and $\log \Delta t_z$ are the Z-transform of $\log \Theta_n$, $\log \phi_n$ and $\log \Delta t_n$, respectively. Thus, an algebraic equation is obtained where the independent variable is z. This

*The Z-transform (also called discrete Laplace transform) is for discrete systems the equivalent to the Laplace transform in continuous systems.

equation can be again represented by means of block diagrams resulting Figure 5.2.

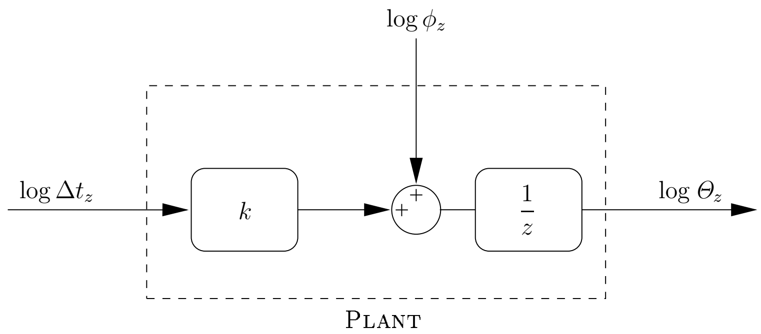


Figure 5.2: Block diagram in the Z-domain.

Obtaining equation (5.10) is the first step in developing adaptive strategies based on control theory. The method of operation of the adaptive algorithm consists of varying the time step size such that the estimated error is close to the tolerance.

5.2.1 Feedback. Closed loop dynamics

In order to control the system depicted in Figure 5.2, it is needed to feedback it, comparing the output of the system to a reference signal. In other words, the goal of the adaptive algorithm is to keep the estimated error as close as possible to the tolerance. For that, the control system computes the difference between these two signals and in function of this difference the system increases or decreases the time step size to fulfill its objective. This idea is represented graphically in Figure 5.3.

The part which adapts the time step size is called *controller*. Denoting the tolerance by $\log \varepsilon_z$, the controller takes the difference $(\log \varepsilon_z - \log \Theta_z)$ and acts to correct the time step size. That is, the controller has as input $(\log \varepsilon_z - \log \Theta_z)$ and as output $\log \Delta t_z$ and thus, the following equation holds:

$$\log \Delta t_z = G_c(z) \cdot (\log \varepsilon_z - \log \Theta_z) \ , \tag{5.11}$$

where $G_c(z)$ is the transfer function of the controller which, for the moment is unknown. The remainder of this chapter studies how this controller must be selected for obtaining an efficient adaptive scheme.

From the point of view of control theory, the input $\log \phi_z$ is seen as a *perturbation.* When the system is stabilized, i.e. $\log \Theta_z = \log \varepsilon_z$, if the signal

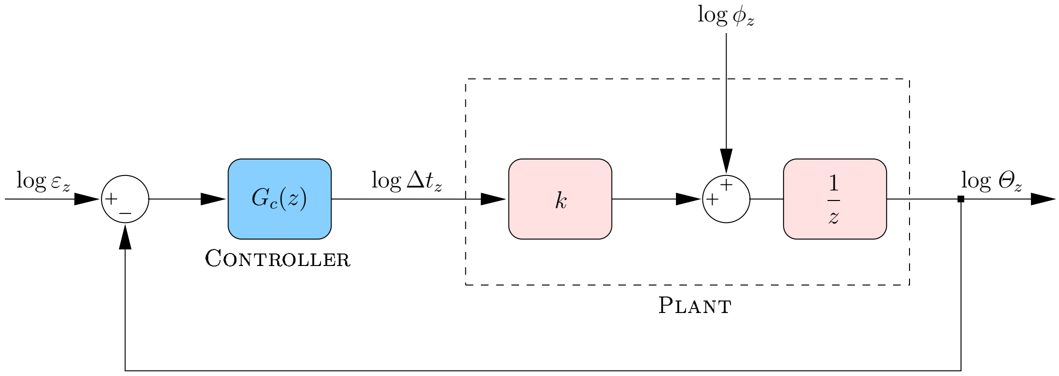


Figure 5.3: Complete block diagram of the time step size control process in closed loop. (In dashed lines it is the open loop system).

$\log \phi_z$ changes its value, then the system is unstabilized and the controller must correct this deviation.

Gathering equations (5.9) and (5.11) the system is perfectly described and thus, the following system of equations is stated:

$$\begin{cases} \log \Theta_z = z^{-1} \log \phi_z + G_p(z) \log \Delta t_z \ , \\ \log \Delta t_z = G_c(z) \cdot (\log \varepsilon_z - \log \Theta_z) \ . \end{cases} \tag{5.12}$$

From system (5.12), the variables $\log \Theta_z$ and $\log \Delta t_z$ can be expressed as function of the inputs $\log \varepsilon_z$ and $\log \phi_z$:

$$\begin{cases} \log \Theta_z = F_\varepsilon(z) \log \varepsilon_z + F_\phi(z) \log \phi_z \ , \\ \log \Delta t_z = H_\varepsilon(z) \log \varepsilon_z + H_\phi(z) \log \phi_z \ . \end{cases} \tag{5.13}$$

where the functions $F_\varepsilon(z)$, $F_\phi(z)$, $H_\varepsilon(z)$ and $H_\phi(z)$ are given by:

$$\begin{aligned} F_\varepsilon(z) &= \frac{G_p(z) \cdot G_c(z)}{1 + G_c(z) \cdot G_p(z)} \ , \qquad & F_\phi(z) &= \frac{z^{-1}}{1 + G_c(z) \cdot G_p(z)} \ , \\ H_\varepsilon(z) &= \frac{G_c(z)}{1 + G_c(z) \cdot G_p(z)} \ , \qquad & H_\phi(z) &= \frac{z^{-1} \cdot G_c(z)}{1 + G_c(z) \cdot G_p(z)} \ . \end{aligned} \tag{5.14}$$

The study of these transfer functions provides us with valuable information for determining an efficient controller. This point is discussed in Section 5.4.

Remark 5.3. If an expression for $G_c(z)$ is given, the from equation $(5.12)_2$, a relation for selecting the time step size can be recovered (see Section 5.3). □

5.2.2 Accepting and rejecting time steps

In this section we discussed when a step is accepted or rejected and also how the tolerance is selected. First, let us consider that a simulation has been carried out up to instant t_n. At this time, an estimate for the error is computed (recall that this estimation is denoted by Θ_n). Also, a tolerance ε_n is defined at this time.

Then, a step is accepted when the following relation holds:

$$\Theta_n \leqslant s_f \cdot \varepsilon_n \ , \tag{5.15}$$

where s_f is a slack factor greater than 1. That is, a step is considered to be sufficiently accurate if the estimated error is smaller than the tolerance multiplied by a security factor. This factor avoid rejecting a step when the error is greater than the tolerance but very close to it, since computing a new step would not be efficient to obtain almost the same error. In this work, we propose a value for s_f of $s_f = 1.05$.

In those cases in which the error is greater than the security factor times the tolerance, the step is rejected and recomputed again using a smaller time step size until condition (5.15) is satisfied.

Throughout this work the estimated error is normalized using the energy norm. Hence, to be consistent the tolerance will be a fraction of the square root of the mechanical energy, i.e.,

$$\varepsilon_n = \gamma \cdot \sqrt{E_n} \ , \tag{5.16}$$

where E_n is the mechanical energy (kinetic energy + deformation energy) of the system and γ is a parameter defined by the user.

In some works such us SÖDERLIND [2002] and SÖDERLIND [2003], among others, the tolerance is kept fixed along the analysis. Here, in contrast, a variable tolerance is considered.

For example, in the numerical simulations presented in Section 5.7 we have used the following values for parameter γ : 0.001, 0.01 and 0.02.

5.3 Types of controllers

In the previous section a model for selecting the time step was built. The only part which was not completely defined was the controller. Here, three of the most used types of controllers are discussed.

5.3.1 Integral (I) controller

The most basic controller is called as Integral controller and it is denoted by I. It has the following transfer function (see, e.g., PHILLIPS & NAGLE [1998]):

$$G_c^{\mathrm{I}}(z) = k_i \frac{z}{z-1} \ . \tag{5.17}$$

By introducing (5.17) in $(5.12)_2$ and performing the inverse Z-transform to the resulting expression, then the following formula is reached:

$$\Delta t_{n+1} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}} \right)^{k_i} \cdot \Delta t_n \tag{5.18}$$

where k_i is a free parameter at this moment.

If $k_i = 1/k$, formula (5.18) becomes the widely used standard algorithm discussed in Section 5.1.

As commented before, this formula has a good performance in smooth problems. However, in non-smooth cases it tries to increase in a great amount the time step size, producing that many steps must be rejected. Also, it has another drawback. Although a step is accepted, as the time step size is larger than with other controllers the Newton-Raphson iterations needed to solve it will be more than for other controllers. These aspects will be shown in the numerical simulations of Section 5.7.

5.3.2 Proportional-Integral (PI) controller

A more sophisticated controller is the Proportional-Integral (PI). It is a generalization of the I controller since it adds a proportional action to the integral one. Hence, its transfer function is

$$G_c^{\mathrm{PI}}(z) = k_p + k_i \frac{z}{z-1} \ . \tag{5.19}$$

This transfer function leads to the following formula:

$$\Delta t_{n+1} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{k_p+k_i} \left(\frac{\varepsilon_n}{\Theta_n}\right)^{-k_p} \cdot \Delta t_n \ . \tag{5.20}$$

It can be observed that this formula takes into account besides the ratio tolerance/error at t_n, the ratio at instant before. If $k_p = 0$, then the I controller is recovered.

5.3.3 Proportional-Integral-Derivative (PID) controller

The most used controller is the Proportional-Integral-Derivative (PID) due to its better performance. It adds a derivative action to the PI controller, resulting

$$G_c^{\mathrm{PID}}(z) = k_p + k_i \frac{z}{z-1} + k_d \frac{z-1}{z} \ . \tag{5.21}$$

The final formula for selecting the time step size is given by

$$\Delta t_{n+1} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{k_p+k_i+k_d} \left(\frac{\varepsilon_n}{\Theta_n}\right)^{-(k_p+2k_d)} \left(\frac{\varepsilon_{n-1}}{\Theta_{n-1}}\right)^{k_d} \Delta t_n \ , \tag{5.22}$$

where as it takes into account information at t_{n+1}, t_n and t_{n-1}, it produces smoother time step sequences than the I controller.

The three controllers discussed can be generalized in the following single formula

$$\Delta t_{n+1} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{\beta_1} \left(\frac{\varepsilon_n}{\Theta_n}\right)^{\beta_2} \left(\frac{\varepsilon_{n-1}}{\Theta_{n-1}}\right)^{\beta_3} \Delta t_n \ , \tag{5.23}$$

where β_i $(i = 1, 2, 3)$ are given in Table 5.1.

Controller	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$
I	k_i	0	0
PI	$k_p + k_i$	$-k_p$	0
PID	$k_p + k_i + k_d$	$-(k_p + 2k_d)$	k_d

Table 5.1: Values of β_1, β_2 and β_3 for the different controllers.

Remark 5.4. Initially, the PI controller needs to have information of two steps (the PID controller needs three). Then, the first and second steps are done using the I controller and from it, the PI controller can already be used.

□

5.4 Choice of parameters for the controllers

A common technique in control theory to study the response of a closed loop system is to investigate its behaviour in the frequency domain. As commented before, the input $\log \phi_n$ is seen as a perturbation that unstabilized the system and the goal of the controller is to minimize its adverse effect.

The range of values that can adopt this perturbation includes from the smoothest input, i.e, a constant sequence to the most non-smooth behaviour which is represented by the sequence $(-1)^n$. By performing the discrete Fourier transform to convert this signal to the frequency domain, the perturbation is represented by the sinusoidal signal "$\cos \omega n$" with $\omega \in [0, \pi]$ (see SÖDERLIND [2002] for details). The case where $\omega = 0$ corresponds to a constant signal whereas the case where $\omega = \pi$ corresponds to the signal $(-1)^n$.

Thinking in a mechanical analysis, a smooth problem has predominance of low values of frequency ω. However, in a non-smooth problem (for example, simulations with intermittent contacts) the perturbation is mainly composed by signals of high frequencies (close to π). Hence, depending of type of problem we will have to eliminate either low frequencies (smooth problems) or the high ones (non-smooth problems).

As defined in (5.13) and (5.14), the influence in the response of the system of the perturbation $\log \phi_z$ is governed by the transfer functions $F_\phi(z)$ and $H_\phi(z)$. By attenuating their amplitudes, the effect of the perturbation will be minimized.

As we are working in the frequency domain, it is usual to represent the behaviour of any transfer function in the so-called Bode diagrams. These diagrams show the amplitude and the phase of a transfer function for a range of frequencies. The frequency ω is usually represented in a logarithm scale and the amplitude is plotted in decibels. This unit is defined as follows.

Given a quantity ϑ, its value in decibels (dB) is given by:

$$20 \log_{10} |\vartheta| \ . \tag{5.24}$$

In this work we concentrate in the amplitude of the different signals and thus, the phase response will not be studied. Figure 5.4 depicts the amplitude of the transfer functions $F_\phi(z)$ (top) and $H_\phi(z)$ (bottom) for different controllers. The analysis is started studying the I controller with a value for its parameter $k_i = 1/k$. The first important fact is that this controller has not effect in the amplitude of $H_\phi(\omega)$. For all range of frequencies the gain is 1 (0 in the log graph). However, respect to the function $F_\phi(\omega)$ the I controller attenuates low frequencies but amplifies the high ones (from $\omega \approx 1.05$). It should be noted that positive values of the amplitude expressed in dB mean amplification whereas negative values mean attenuation. Clearly, this standard controller must be chosen only for smooth problems since in non-smooth simulations it is not capable of attenuating the perturbations.

Now, the goal is to find good values for the parameters of PI and PID controllers which improve the performance of the I controller in non-smooth problems.

In Söderlind [2003] it is demonstrated that there exists a complementarity condition between $F_\phi(\omega)$ and $H_\phi(\omega)$. That is, it is not possible to attenuate, in an important percentage, the response of both transfer functions for the same frequency. Thus, a compromise solution must be adopted.

With this in mind the three following controllers are proposed and studied:

i. PI controller with $\beta_1 = \beta_2 = 1/(6k)$,

ii. PID controller with $\beta_1 = \beta_3 = 1/(18k)$ and $\beta_2 = 1/(9k)$ (it will be referred to as PID-a),

iii. PID controller with $\beta_1 = \beta_3 = 1/(30k)$ and $\beta_2 = 1/(15k)$ (it will be referred to as PID-b),

taking into account that $k = p + 1$, being p the order or accuracy of the time-stepping method.

Concentrating in high frequencies, the three controllers compared provide an attenuation for function $H_\phi(\omega)$ (see the graph on the bottom of Figure 5.4). More specifically, the PID-b controller is the one that has more attenuation at high frequencies.

With respect to the transfer function $F_\phi(\omega)$, all of them provide a response very close to zero (neither amplification nor attenuation).

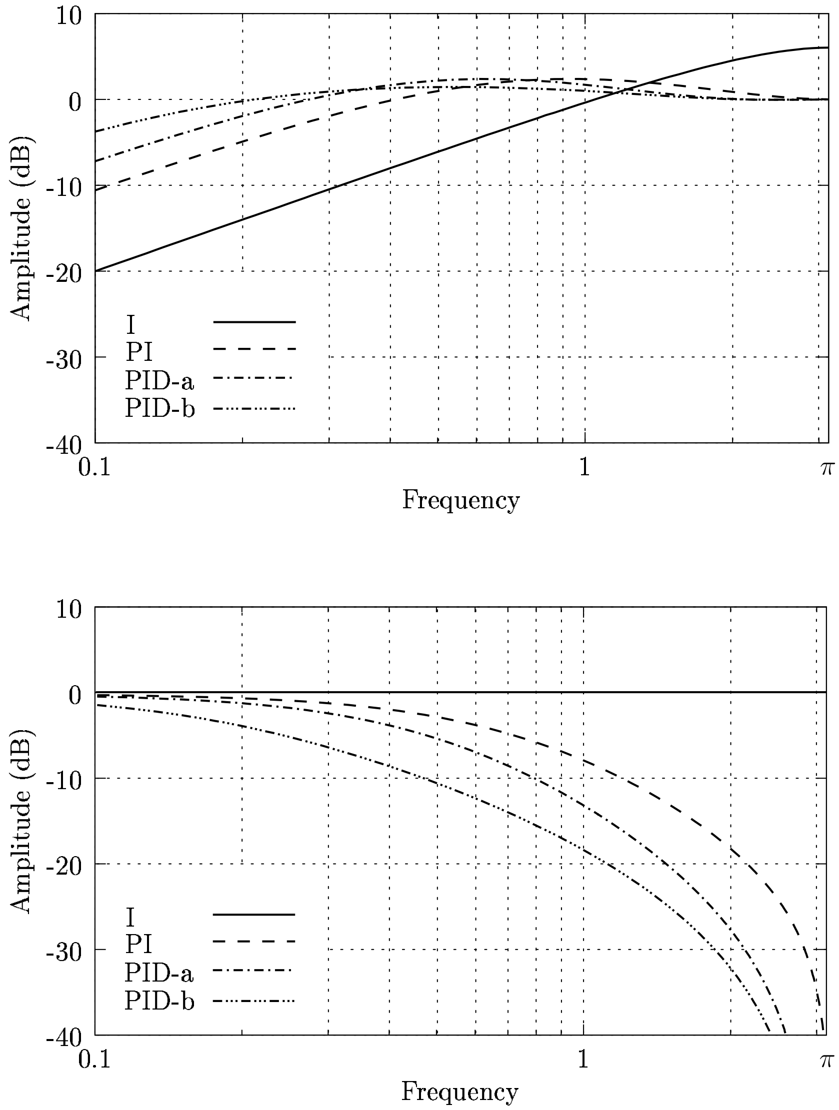


Figure 5.4: Closed loop response in the frequency domain. Top: amplitude of the transfer function $F_\phi(\omega)$; Bottom: amplitude of the transfer function $H_\phi(\omega)$. The parameters for the controllers are: $\beta_1 = 1/k$ for I controller; $\beta_1 = \beta_2 = 1/(6k)$for PI controller; $\beta_1 = \beta_3 = 1/(18k)$ and $\beta_2 = 1/(9k)$ for PID-a controller; $\beta_1 = \beta_3 = 1/(30k)$ and $\beta_2 = 1/(15k)$ for PID-b controller)

All these conclusions will be ratified in different numerical simulations presented in Section 5.7, where it will be shown as the PID-b controller is very efficient in non-smooth dynamic analyses.

5.5 An improved adaptive strategy for selecting the time step size based on control theory

The model developed in the previous sections is based on that the local error fulfil the expression:

$$\Theta_{n+1} = \phi_n \Delta t_n^k \ , \qquad \text{with } k = p+1, \tag{5.25}$$

where p is the order of accuracy of the time-stepping method.

This expression is valid in the asymptotic regime, that is, in the limit when $\Delta t_n \to 0$. Thus, for any finite value of the time step Δt_n this theoretical asymptotic behaviour cannot hold. Also, when forces which act over a deformable solid are non-smooth this ideal order of convergence can be degraded.

We propose to take this fact into account with the object of introducing an improvement in the adaptive strategy. The general idea is as follows. First, we assume that expression (5.25) is fulfilled, i.e., it is supposed that the time-stepping method exhibits a order of accuracy p. If a change in the behaviour of the method is detected, then the order of accuracy is reduced to $p-1$.

This change can be detected by following the evolution of the perturbation ϕ_n. Let σ be the relative variation between two consecutive value of ϕ, i.e,

$$\sigma = \frac{|\phi_{n+1} - \phi_n|}{\max(\phi_n, \phi_{n+1})} \ . \tag{5.26}$$

Then, when σ exceeds a predefined threshold, a change is detected and the order of the method is decreased. The quantity ϕ_n is computed from expression (5.8):

$$\phi_n = \frac{\Theta_{n+1}}{\Delta t_n^{p_n+1}} \ , \tag{5.27}$$

where $p_n = p$ if the previous step was "smooth" or $p_n = p-1$ if an abrupt change was detected in the last step.

Those strategies that use the improvement described in this section will be referred to as I-i, PI-i and PID-i according to they employ the I, PI or PID controller, respectively.

5.6 Implementation of the adaptive algorithm

In this section we summarize the final implementation of the adaptive strategy proposed.

Given the solution at t_n, compute the solution at t_{n+1}.

1. Estimate the local error Θ_{n+1}.

2. Compare the estimated local error to the tolerance:

2.a. `IF` $\Theta_{n+1} < s_f \cdot \varepsilon_{n+1} \Longrightarrow$ Accept time step $\Longrightarrow$ `GO TO` Box 5.2

2.b. `IF` $\Theta_{n+1} > s_f \cdot \varepsilon_{n+1} \Longrightarrow$ Reject time step $\Longrightarrow$ `GO TO` Box 5.3

Box 5.1: Compare the estimated local error to the tolerance.

Advance to the following step with a time step size Δt_{n+1} such that:

1. `IF` an *abrupt change* was detected, then

$$\Delta t_{n+1} = \Delta t_n \; , \tag{5.28}$$

`ELSE`

$$\Delta t_{n+1} = \begin{cases} \left(\dfrac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{\beta_1} \cdot \Delta t_n \text{ for I controller} \\[2ex] \left(\dfrac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{\beta_1} \cdot \left(\dfrac{\varepsilon_{n}}{\Theta_{n}}\right)^{\beta_2} \cdot \Delta t_n \text{ for PI controller} \\[2ex] \left(\dfrac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{\beta_1} \cdot \left(\dfrac{\varepsilon_{n}}{\Theta_{n}}\right)^{\beta_2} \cdot \left(\dfrac{\varepsilon_{n-1}}{\Theta_{n-1}}\right)^{\beta_3} \cdot \Delta t_n \text{ for PID controller} \end{cases} \tag{5.29}$$

Box 5.2: Advancing a step.

Recompute the step with a smaller time step size.

1. Compute deviation σ:

$$\phi_n = \frac{\Theta_n}{\Delta t_{n-1}^{p_n+1}} \tag{5.30}$$

$$\phi_{n+1} = \frac{\Theta_{n+1}}{\Delta t_n^{p+1}} \tag{5.31}$$

$$\sigma = \frac{|\phi_{n+1} - \phi_n|}{\max(\phi_n, \phi_{n+1})} \tag{5.32}$$

2. Decrease the time step size:

2.a. `IF` $\sigma > 0.50 \Longrightarrow$ *abrupt change* detected $\Longrightarrow$

$$\Delta t_n^{\text{new}} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{1/p} \cdot \Delta t_n^{\text{old}} \ . \tag{5.33}$$

$$p_n = p - 1 \ . \tag{5.34}$$

2.b. `IF` $\sigma < 0.50 \Longrightarrow$ Smooth step $\Longrightarrow$

$$\Delta t_n^{\text{new}} = \left(\frac{\varepsilon_{n+1}}{\Theta_{n+1}}\right)^{1/(p+1)} \cdot \Delta t_n^{\text{old}} \ . \tag{5.35}$$

$$p_n = p \ . \tag{5.36}$$

Box 5.3: Decreasing the time step size.

Remarks 5.5.

i. As can be observed from the implementation scheme given in the previous boxes, the computational cost of changing the time step size is negligible.

i. In the boxes it is described the improved strategy (I-i, PI-i and PID-i). The normal algorithms (I, PI and PID) follow the same scheme but keeping constant the order p of the time-stepping method and not detecting the abrupt changes in the perturbation ϕ.

iii. Whereas the I controller works well in smooth problems, we are going to show as the i strategies (I-i, PI-i and PID-i with parameters given in Section 5.4) are more efficient in non-smooth simulations, confirming the theoretical results.

□

5.7 Numerical simulations

This section analyzes the performance of the different adaptive strategies presented in the previous sections through three numerical simulations.

All examples are non-linear dynamic analyses and they are presented in an increasing order of complexity. The first one study the behaviour of a cantilever beam subjected to a non-smooth loading function. The second example simulates the evolution of a deformable ball impacting against a rigid plane. Finally, the last example is a more real problem. It is the impact of a deformable dummy against an airbag which is deploying.

In all cases the error estimator used is the one described in Chapter 4 of this dissertation.

5.7.1 Example 1: Non-linear elastic cantilever beam

The first numerical simulation is the dynamic analysis of a cantilever beam. The material of the beam is non-linear elastic with neo-Hookean hyperelastic response and the following parameters: Young's modulus $E = 10^7$ MPa, Poisson's ratio $\nu = 0.3$ and density $\rho = 1000$ kg/m^3.

An extreme of the beam is clamped whereas in the other one there exists a vertical distributed load q as depicted in Figure 5.5. Also, in this figure, the geometry and dimensions of the beam are given. The distributed load varies in time according to a periodic triangular function of amplitude $q = 10000$ N/m and period 0.4 s and the body forces are not taken into account.

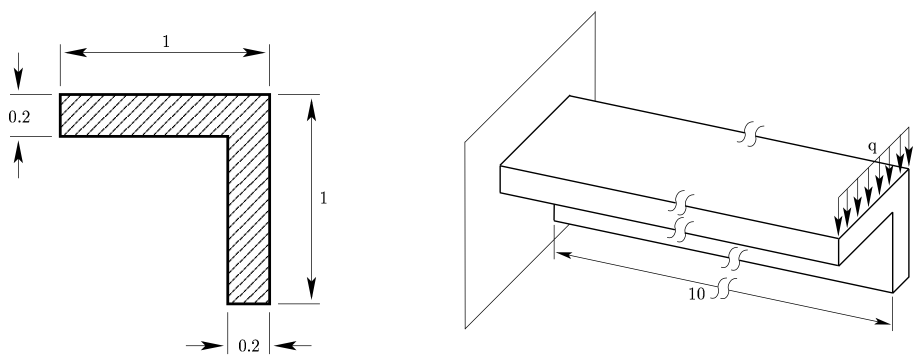


Figure 5.5: Hyperelastic cantilever beam. Geometry of the model with dimensions expressed in meters.

The finite element mesh built is perfectly regular and has 5757 nodes and 3600 brick elements with a standard finite deformation formulation. All elements are equal of size $0.10 \times 0.10 \times 0.10$ m^3. The whole model has a total of 17100 degrees of freedom.

The analysis is carried out from time 0 to 2 s starting with an initial time step $\Delta t_0 = 0.025$ s and using the HHT method with $\alpha = 0.95$.

We solve each step such that the estimated error is smaller than 0.01 times the square root of the mechanical energy of the system, as commented in Section 5.2.2. Six different strategies are compared: I, I-i, PI, PI-i, PID and PID-i. The value of parameters for I controller (and I-i) is $\beta_1 = 1/k$, for the PI controller (and PI-i) are $\beta_1 = \beta_2 = 1/(6k)$ and for the PID (and PID-i) $\beta_1 = \beta_3 = 1/(30k)$ and $\beta_2 = 1/(15k)$. (Initially $k = 3$. recalling that in I-i, PI-i and PI-i strategies, it can be reduced).

Before analyzing the behaviour of the controllers, we show the motion of the beam in Figures 5.6 and 5.7. A snapshot of the deformed position (deformations are not scaled) is depicted each 0.1 s. Also, a contour map of the vertical displacement (in the load direction) is plotted.

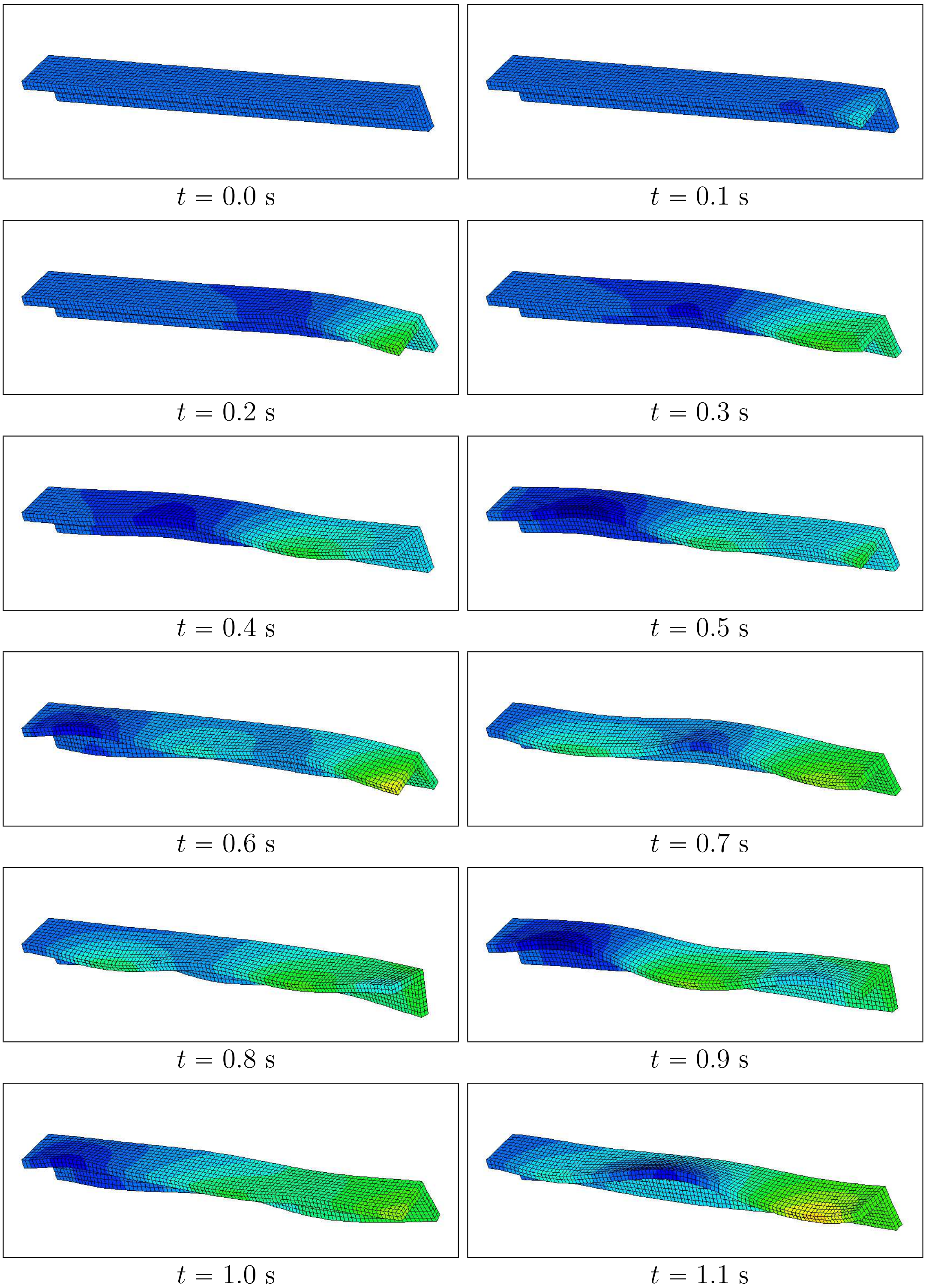
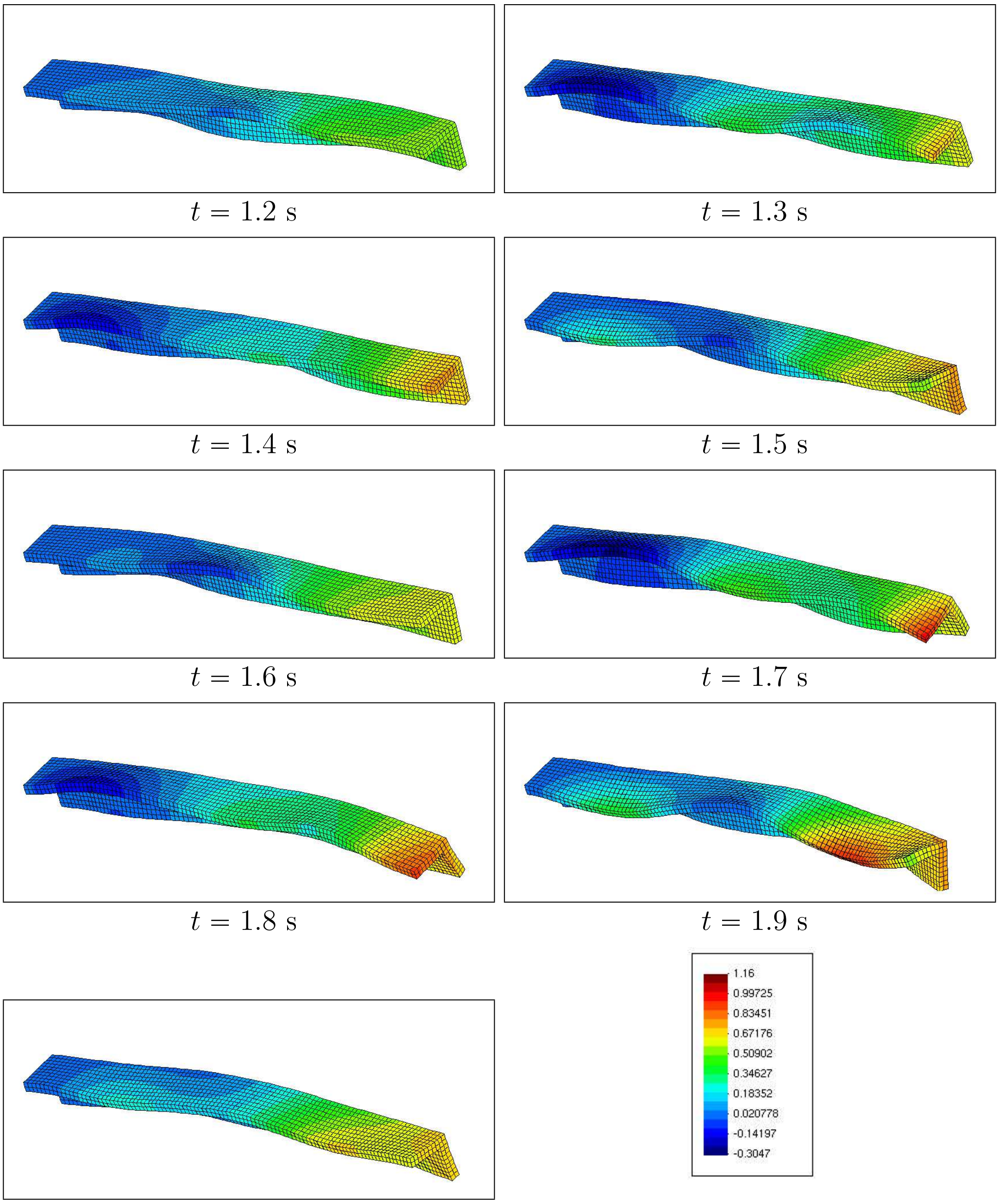


Figure 5.6: Hyperelastic cantilever beam. From top to bottom, left to right, snapshots of the motion plotting the vertical displacement shown every 0.1 s starting at $t = 0$ s (real deformation).



$t = 2.0$ s

Figure 5.7: Hyperelastic cantilever beam. From top to bottom, left to right, snapshots of the motion plotting the vertical displacement shown every 0.1 s starting at $t = 1.2$ s (real deformation).

Now, we present the results obtained with the different adaptive strategies. Table 5.2 summarizes the accepted, rejected and total (accepted+rejected) time steps for the six adaptive strategies compared. It also includes the ratio of the rejected steps respect to the total. It can be observed the difference between the number of the rejected steps for the I controller (141 steps) and for the PID-i (43 steps). This fact results in that in the first case the adaptive algorithm has to reject the 28.54 % of the steps, whereas in the last one only rejects the 9.86 %.

	Time steps			
Controller	**Accepted**	**Rejected**	**Total**	**R/T** (%)
I	353	141	494	28.54
I-i	365	95	460	20.65
PI	381	101	482	20.95
PI-i	371	62	433	14.32
PID	391	72	463	15.55
PID-i	393	43	436	9.86

Table 5.2: Hyperelastic cantilever beam. Comparison of the accepted and rejected steps for the six strategies. The last column (R/T) is the ratio in percentage between the rejected steps and the total ones.

Also, it can be noted that the improved strategies (I-i, PI-i & PID-i) reject fewer steps than their normal strategies (I, PI & PID), respectively.

In Table 5.3 it is given the total number of time steps for each strategy and the number of Newton-Raphson iterations needed to carried out the simulation. Clearly, all strategies improved to the standard rule (I controller). If the scheme based on the PID controller is used, a reduction of 7.16 % is achieved in the number of time steps and of 11.25 % in the iterations. Using the PID-i scheme the results are still better, obtaining a saving in the computational cost of more than a 17 %. In all cases the improvement in the number of iterations is greater than those in the number of time steps.

Figure 5.8 shows the evolution of the number of rejected steps for the I, PID and PID-i controllers. From the first instant, the PID-i scheme rejects much fewer steps than the others two.

Controller	Time steps	Improvement (%)	Iterations	Improvement (%)
I	494	-	1829	-
I-i	460	6.88	1691	7.55
PI	482	2.61	1730	5.85
PI-i	433	12.66	1573	14.80
PID	463	7.16	1652	11.25
PID-i	436	12.53	1536	17.74

Table 5.3: Hyperelastic cantilever beam. Comparison of the total time steps and the Newton-Raphson iterations needed for the six strategies. The improvement is computed respect to the I-controller.

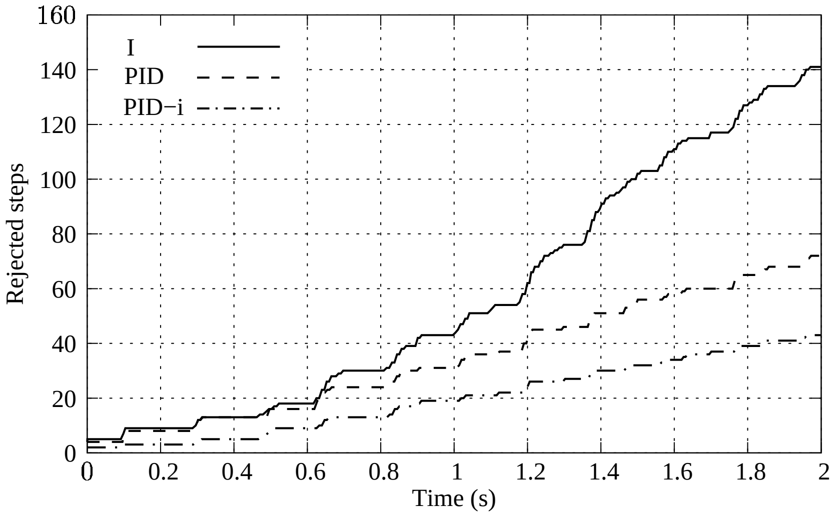


Figure 5.8: Hyperelastic cantilever beam. Evolution in time of the rejected steps for I, PID and PID-i controllers.

5.7.2 Example 2: Hollow ball bouncing over a rigid plane

The second example consists of a hollow ball bouncing on a rigid surface. In Figure 5.9 a section of the geometry of the ball is given. It should be noted that the ball has a cylindrical hole which crosses it.

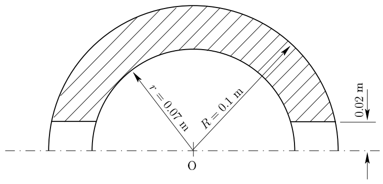


Figure 5.9: Deformable hollow ball. Section with the geometry.

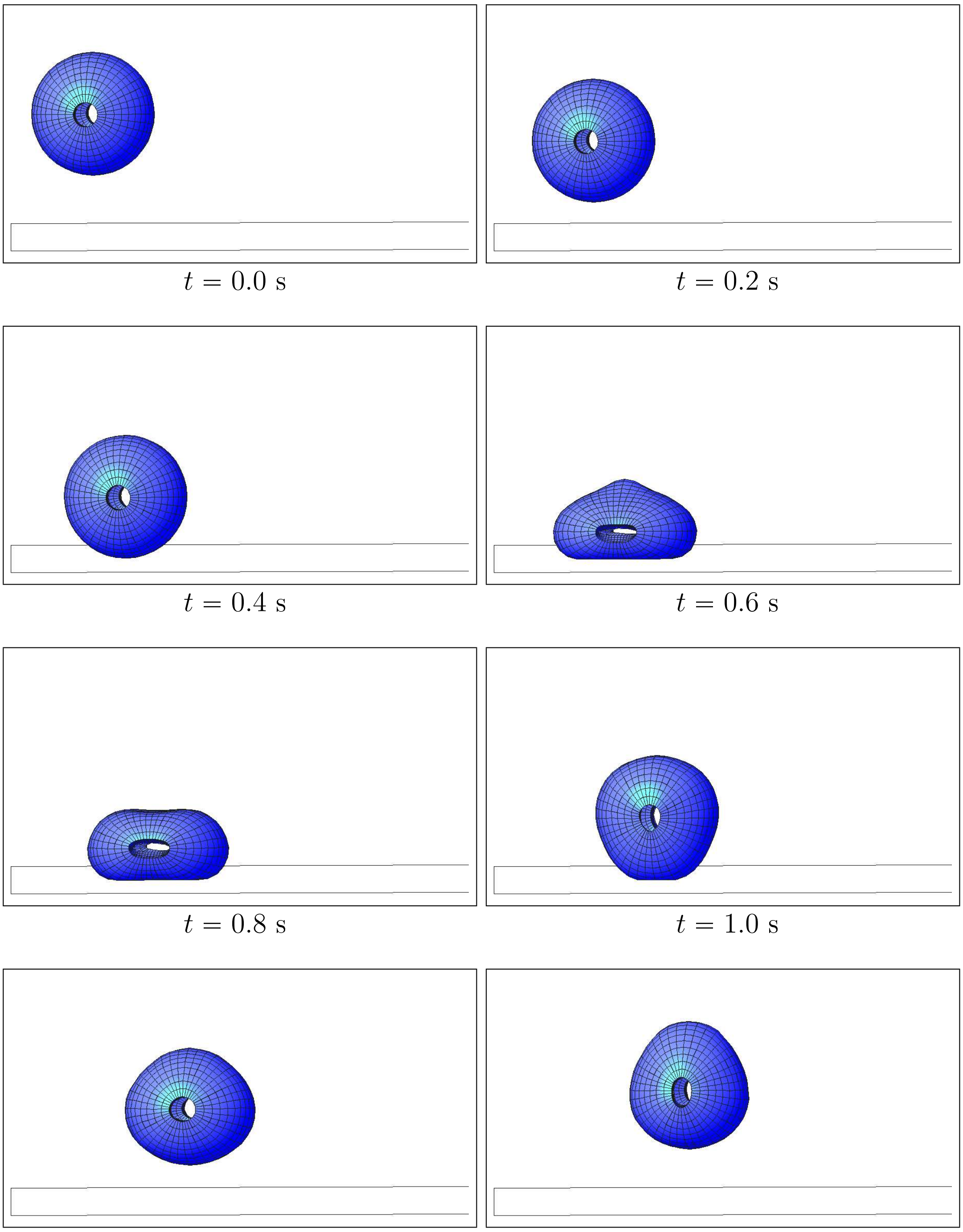
The material of the hollow ball is modeled using a hyperelastic model, specifically a neo-Hookean model with the following parameters: Young's modulus $E = 7 \cdot 10^3$ Pa, Poisson's ratio $\nu = 0.40$ and density $\rho = 2000$ kg/m^3.

The finite element model has 4200 nodes and 3200 brick elements (8-nodes finite deformation hexahedra) with a total of 12600 degrees of freedom and the mesh can be seen in Figure 5.10.

Figure 5.10: Deformable hollow ball. Finite element mesh (Left: whole model; Right: a half of the model).

The ball is initially placed at 0.2 m from the horizontal plane (distance from the center of the ball to the plane) and is released with a velocity of (0.13229,0,-0.12) m/s (see the axis orientation in Figure 5.10). Over the ball acts a gravitational field equal to $g/20$, g being the gravity constant. The contact condition is imposed by a penalty method.

The simulation is performed during three seconds and using the HHT method with $\alpha = 0.95$. In this time, the ball bounces twice against the plane as can be observed in Figures 5.11 and 5.12.



$t = 0.0$ s | $t = 0.2$ s | $t = 0.4$ s | $t = 0.6$ s | $t = 0.8$ s | $t = 1.0$ s | $t = 1.2$ s | $t = 1.4$ s

Figure 5.11: Deformable hollow ball. Solution obtained with the HHT method ($\alpha = 0.95$). From top to bottom, left to right, snapshots of the deformed positions starting at $t = 0$ s and shown every 0.2 seconds (real deformation).

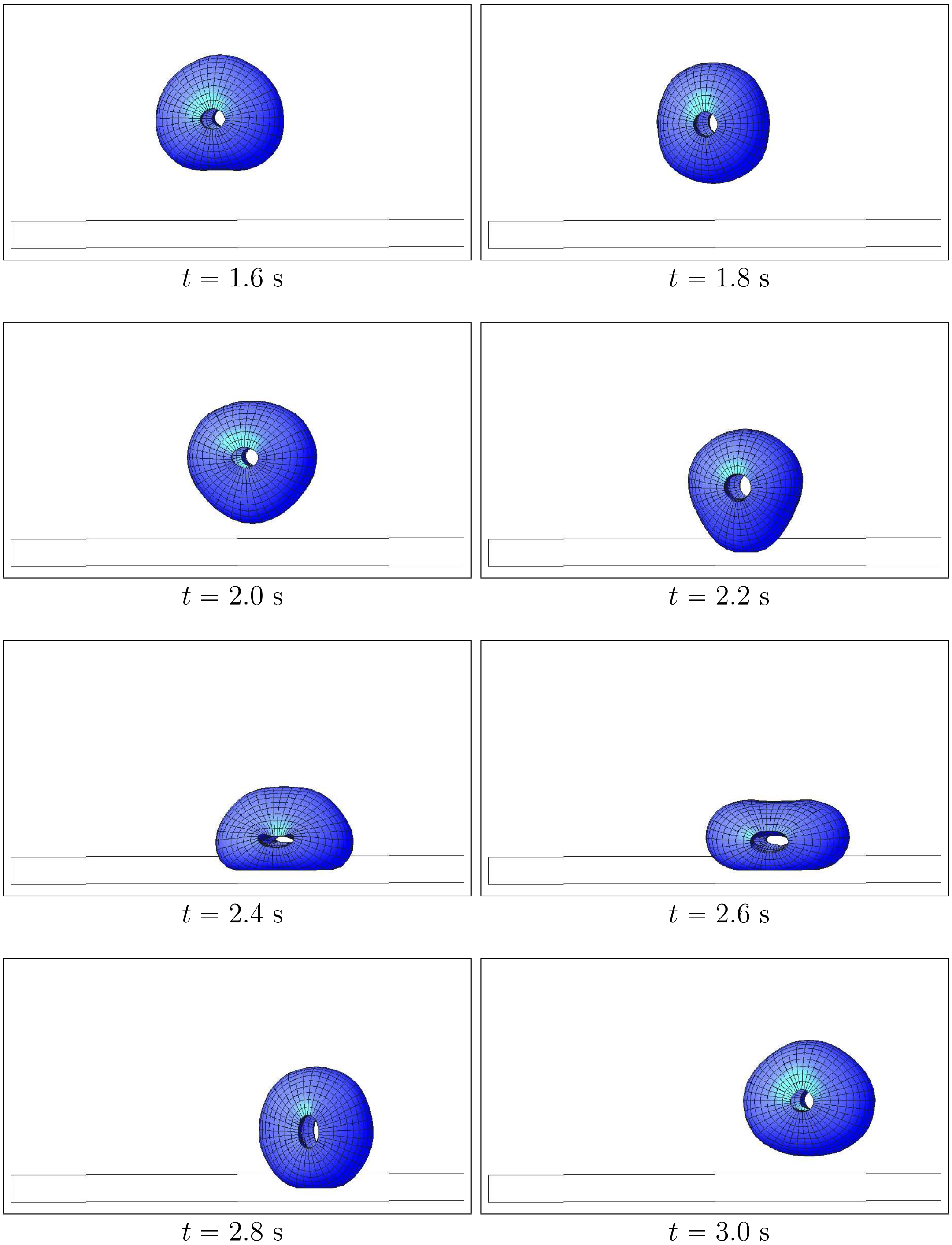


Figure 5.12: Deformable hollow ball. Solution obtained with the HHT method ($\alpha = 0.95$). From top to bottom, left to right, snapshots of the deformed positions starting at $t = 1.6$ s and shown every 0.2 seconds (real deformation).

Now, in this example we adopt the value 0.001 for γ and we analyze five controllers. They are the I, PI, PI-i, PID and PID-i with the same values of their parameters as in the first simulation.

Following the same scheme as in the previous section, we first compare in Table 5.4 the number of accepted and rejected steps of each controller. The I controller rejects 842 steps (15.78 % of total) whereas the PID-i only rejects 182 (3.81 % of total).

	Time steps			
Controller	**Accepted**	**Rejected**	**Total**	**R/T** (%)
I	4493	842	5335	15.78
PI	4502	419	4921	8.51
PI-i	4566	336	4902	6.85
PID	4550	248	4798	5.17
PID-i	4592	182	4774	3.81

Table 5.4: Deformable hollow ball. Comparison of the accepted and rejected steps for the five strategies. The last column (R/T) is the ratio in percentage between the rejected steps and the total ones.

In Table 5.5, it can be appreciated the improvement (10.51 %) obtained in the number of total steps using the PID-i strategy. Regarding the number of iterations, this improvement is a little better (11.38 %). Also, it should be noted that the i strategies (PI-i and PID-i) are slightly more efficient than their respective original controllers (PI and PID).

Controller	**Time steps**	**Improvement** (%)	**Iterations**	**Improvement** (%)
I	5335	-	16887	-
PI	4921	7.76	15526	8.06
PI-i	4902	8.11	15411	8.74
PID	4798	10.06	15064	10.79
PID-i	4774	10.51	14965	11.38

Table 5.5: Deformable hollow ball. Comparison of the total time steps and the Newton-Raphson iterations needed for the five strategies. The improvement is computed respect to the I-controller.

Now, we focus on the evolution of the rejected steps for the worst strategy (I) and for the two best (PID and PID-i). They are depicted in Figure 5.13.

By comparing this figure with those representing the evolution of the hollow ball (Figures 5.11 and 5.12), it is interesting to remark how the rejections are produced when the ball impact against the rigid plane or in other words, when the problem becomes non-smooth.

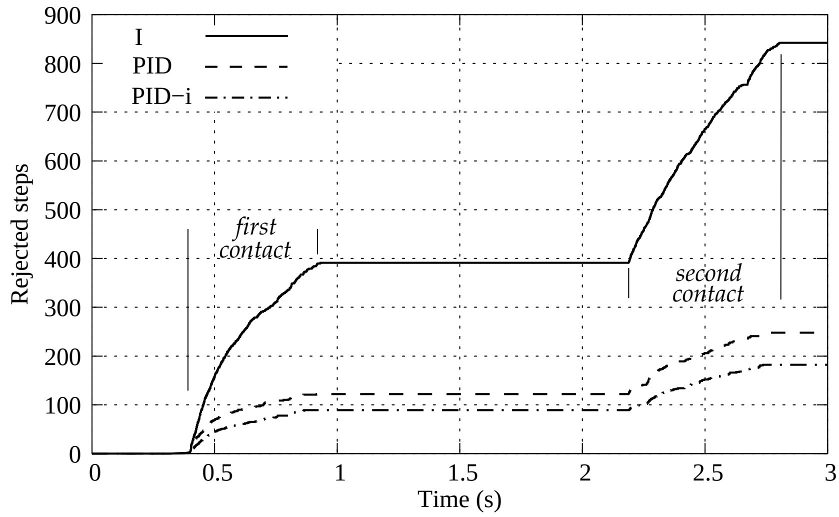


Figure 5.13: Deformable hollow ball. Evolution in time of the rejected steps for I, PID and PID-i controllers. It is pointed out the two time intervals when the contact takes place.

In Figure 5.14 it is shown the instants when an abrupt change is detected by the PID-i strategy. These instants match perfectly with the intervals where the hollow ball collides with the plane.

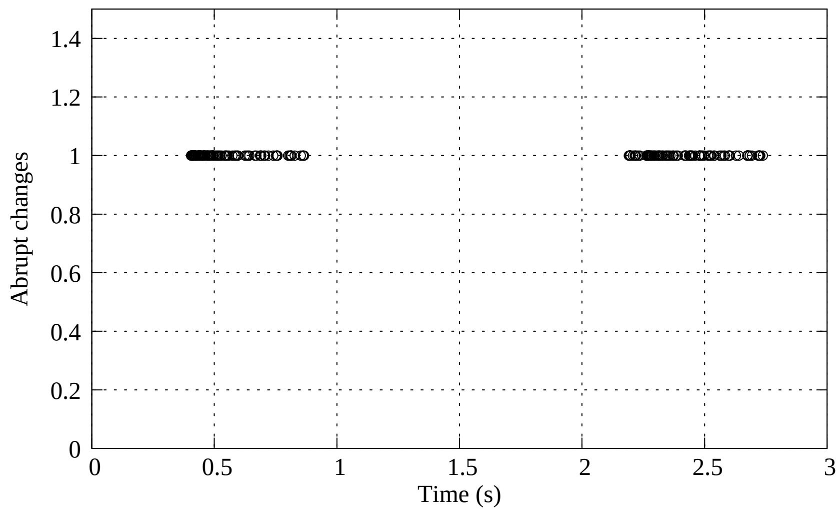


Figure 5.14: Deformable hollow ball. Circles indicate the instant where an abrupt change is detected.

Figure 5.15 depicts the evolution of the time step size during the time interval $0.6 - 0.9$ s (during the first contact). It is observed as the PID-i controller produces a smoother sequence than the I controller which presents a nervous behaviour.

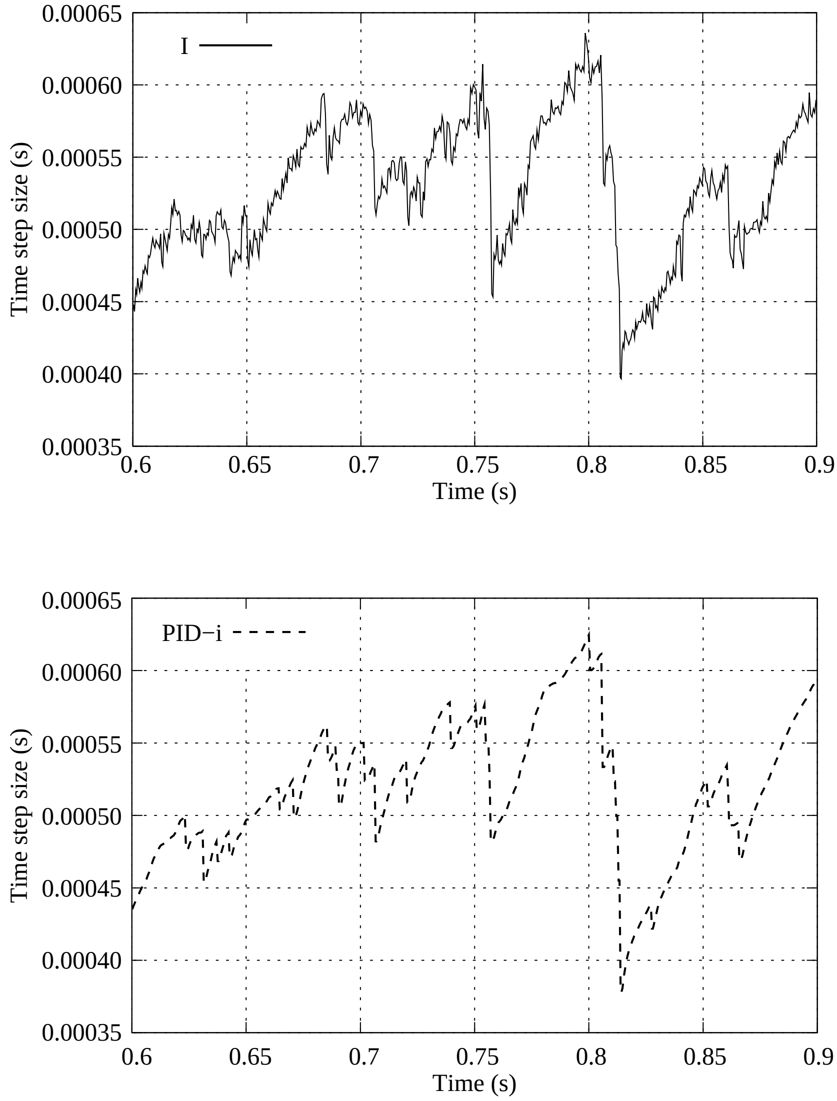


Figure 5.15: Deformable hollow ball. Evolution in time of the time step size between 0.6 s and 0.9 s. (Top: I controller. Bottom: PID-i controller).

5.7.3 Example 3: Deformable dummy impacting against an airbag

The last numerical example of this dissertation is the simulation of the impact of a deformable dummy against an inflating airbag.

For this purpose, a finite model of a dummy and the airbag have been built. Regarding the dummy, it is modeled the superior part of the body, that is, from waist to head, including both arms and neck. The airbag is initially deflated and it is supposed to be circular of radius 0.35 m.

The dummy mesh is composed of 2641 nodes and 9909 elements (tetrahedrons) and the airbag mesh of 4256 nodes and 8468 membrane elements (triangles). The total degrees of freedom are 20520.

The neck is assumed to be a hundred times more flexible than rest of the body being both modelled by means of an Neo-Hookean material. The Young's modulus for the body is $E = 1000$ MPa. The material of the airbag is elastic with Young's modulus $E = 0.5$ MPa and Poisson's ratio $\nu = 0.3$.

The dummy is thrown with speed 5 m/s against an airbag in the inflating process. This inflating process is achieved imposing a internal pressure inside the airbag and the impact between the dummy and the airbag is imposed by a penalty method.

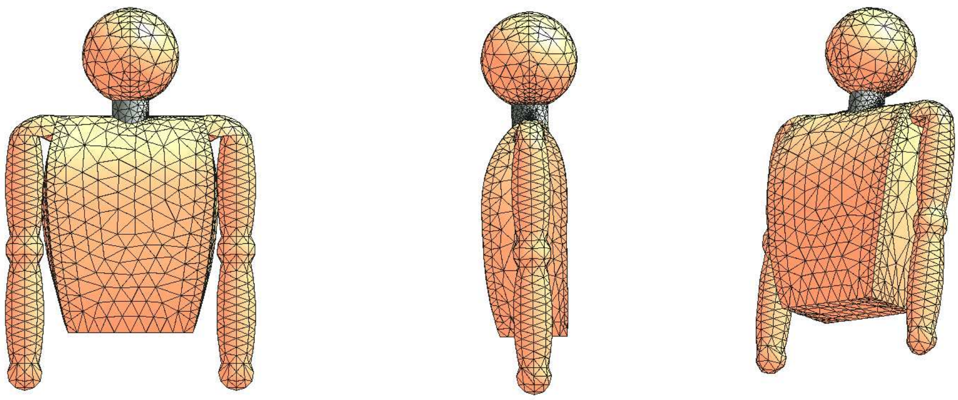


Figure 5.16: Deformable dummy impacting against an airbag. Several views of the dummy.

Figures 5.17, 5.18 and 5.19 depicts the evolution of the airbag and dummy movement observing that the contact between them start between $t = 0.020$ s and $t = 0.025$ s.

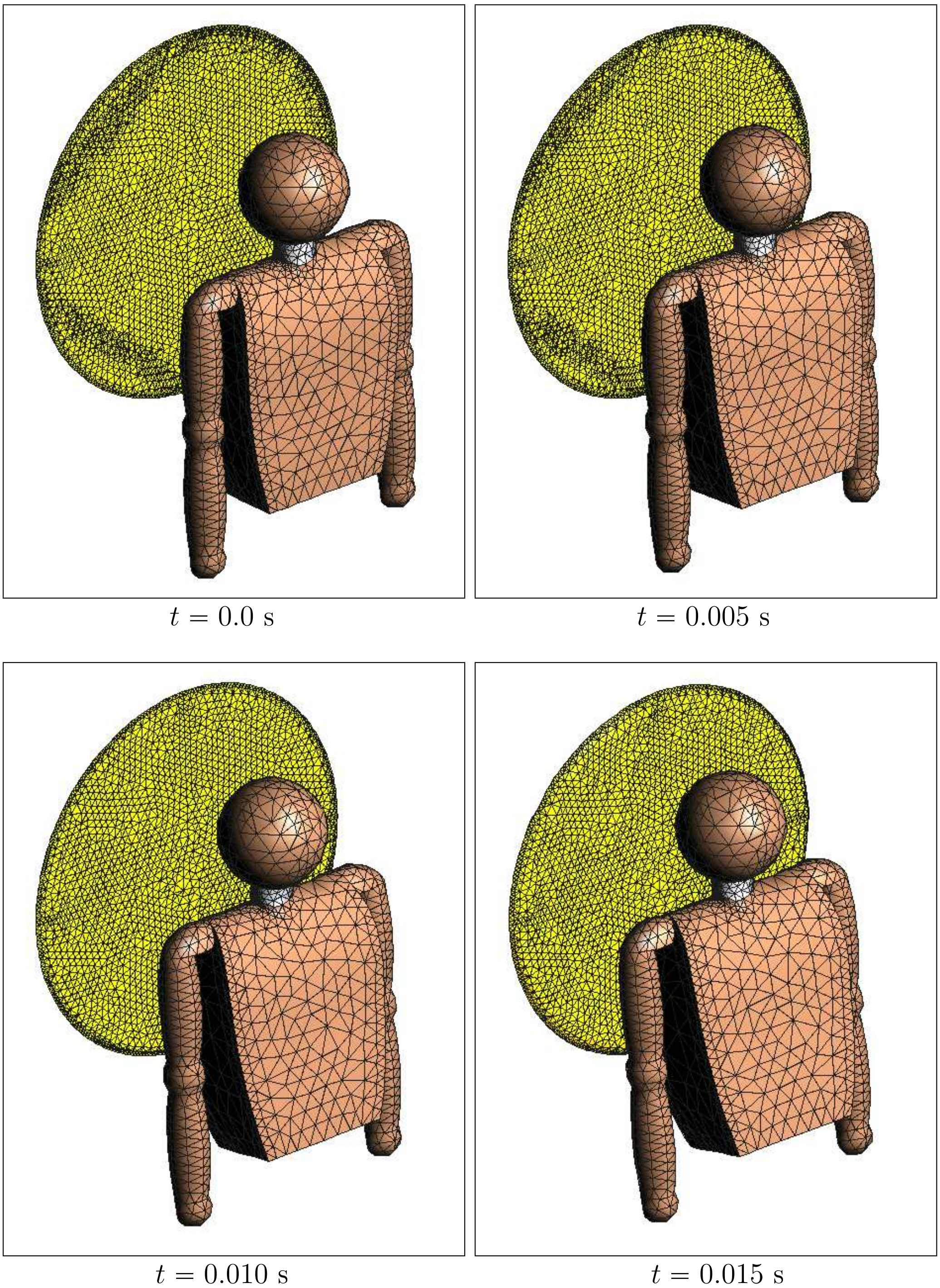
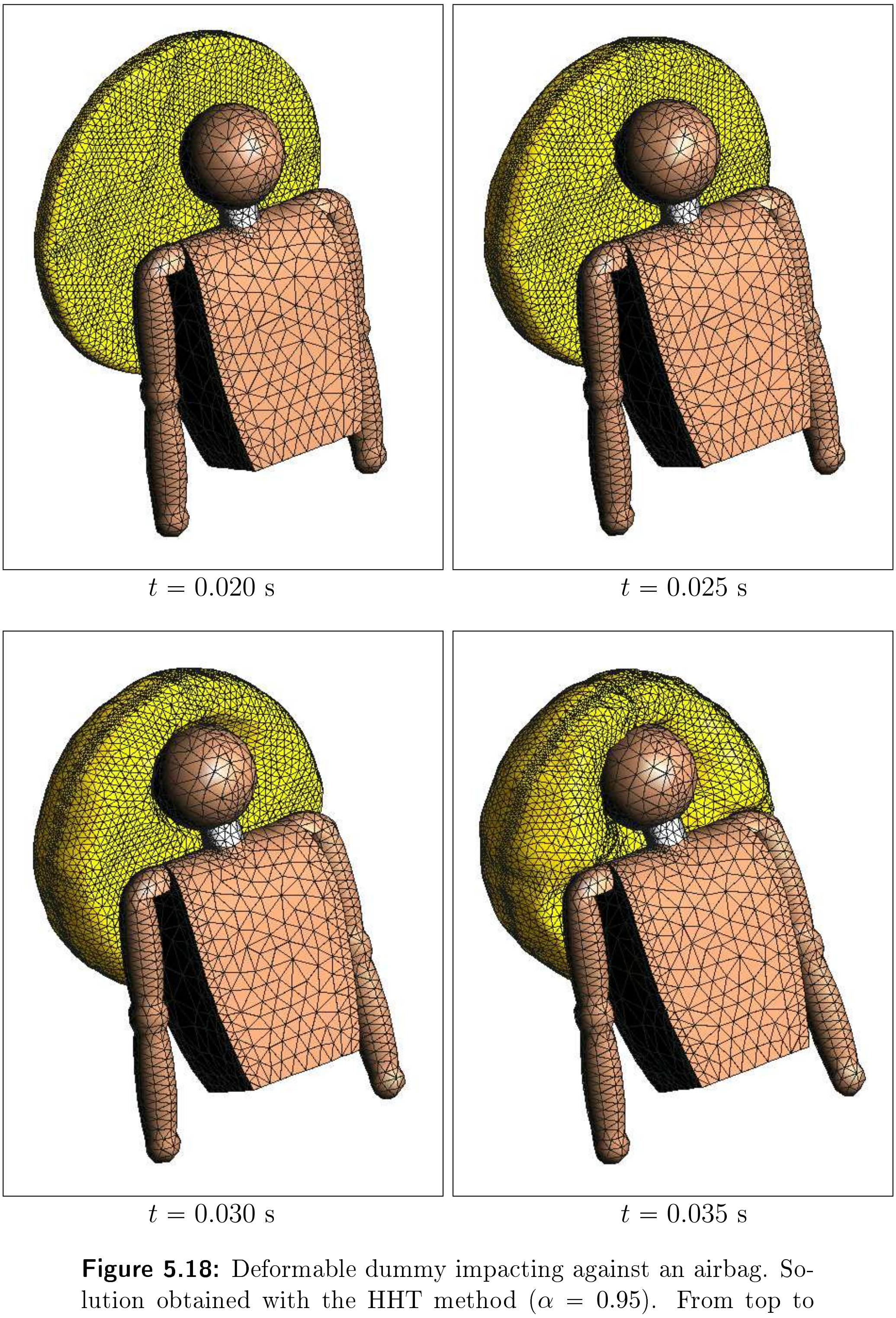


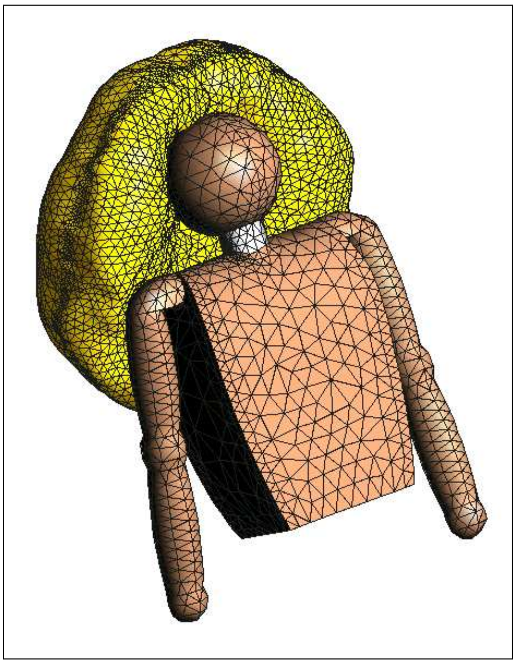
Figure 5.17: Deformable dummy impacting against an airbag. Solution obtained with the HHT method ($\alpha = 0.95$). From top to bottom, left to right, snapshots of the deformed positions starting at $t = 0.0$ s and shown every 0.005 seconds (real deformation).



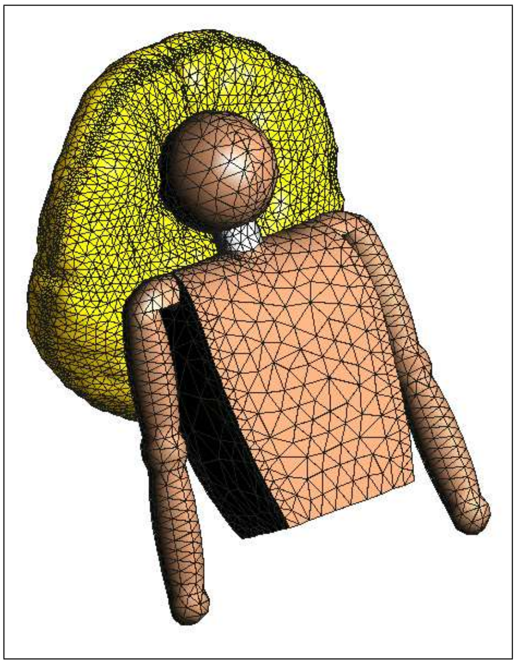
$t = 0.020$ s $\quad$ $t = 0.025$ s

$t = 0.030$ s $\quad$ $t = 0.035$ s

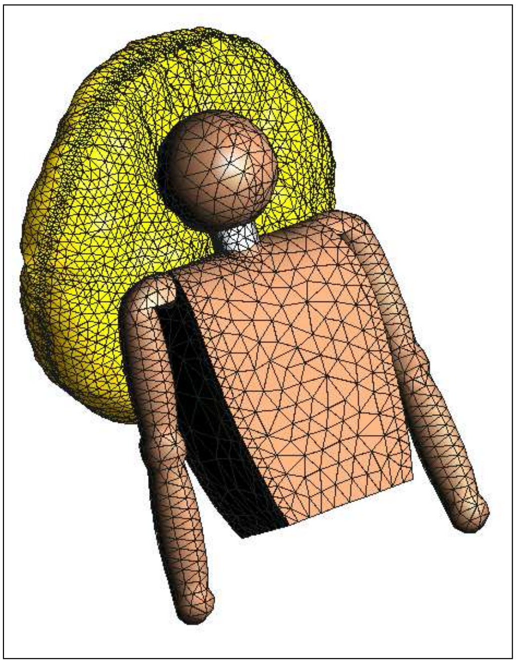
Figure 5.18: Deformable dummy impacting against an airbag. Solution obtained with the HHT method ($\alpha = 0.95$). From top to bottom, left to right, snapshots of the deformed positions starting at $t = 0.020$ s and shown every 0.005 seconds (real deformation).



$t = 0.040$ s



$t = 0.045$ s



$t = 0.050$ s

Figure 5.19: Deformable dummy impacting against an airbag. Solution obtained with the HHT method ($\alpha = 0.95$). From top to bottom, left to right, snapshots of the deformed positions starting at $t = 0.040$ s and shown every 0.005 seconds (real deformation).

The analysis has been performed using the HHT method ($\alpha = 0.95$) and a tolerance $\gamma = 0.02$ and four strategies are analyzed (I, PI, PID and PID-i).

In this more complex example, the conclusions are the same as in the two previous simulations. The PID-i strategy produces fewer rejected steps than the others as can be seen in Table 5.6.

	Time steps			
Controller	**Accepted**	**Rejected**	**Total**	**R/T** (%)
I	1223	469	1692	27.72
PI	1343	298	1641	18.16
PID	1378	180	1558	11.55
PID-i	1367	135	1502	8.98

Table 5.6: Deformable dummy impacting against an airbag. Comparison of the accepted and rejected steps for the four strategies. The last column (R/T) is the ratio in percentage between the rejected steps and the total ones.

Table 5.7 summarizes the total number of time steps and the Newton-Raphson iterations. It is stood out that the improved strategy (PID-i) reduces the computational cost in a 14.10 % respect to the standard scheme (I controller) and also is a 5 % more efficient than the normal one (PID).

Controller	**Time steps**	**Improvement** (%)	**Iterations**	**Improvement** (%)
I	1692	-	6734	-
PI	1641	3.01	6530	3.02
PID	1558	7.92	6090	9.56
PID-i	1502	11.23	5785	14.10

Table 5.7: Deformable dummy impacting against an airbag. Comparison of the total time steps and the Newton-Raphson iterations needed for the four strategies. The improvement is computed respect to the I-controller.

The rejected steps start to increase dramatically when the dummy impacts the airbag, as can be observed in Figure 5.20. The PID-i strategy captures perfectly the interval of contact between the dummy and the airbag (from $t = 0.022$ s to $t = 0.05$ s). It is shown in Figure 5.21. In this interval the problem becomes non-smooth and is where the PID-i strategy has a better performance.

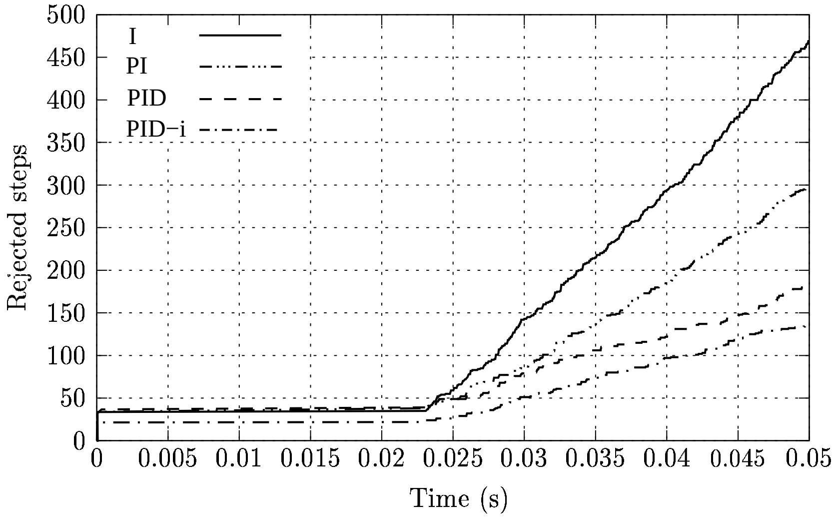


Figure 5.20: Deformable dummy impacting against an airbag. Evolution in time of the rejected steps for I, PI, PID and PID-i controllers.

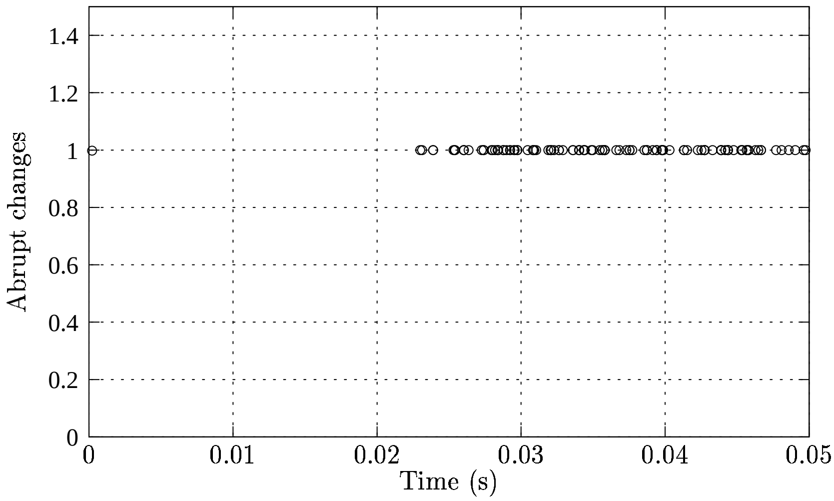


Figure 5.21: Deformable dummy impacting against an airbag. Circles indicate the instant where an abrupt change is detected

Finally, we see that the PID-i scheme computes smoother sequences of the time step size than the standard I controller as is shown Figure 5.22.

We have shown in this section how in non-smooth problems, the PI, PI-i, PID and above all the PID-i strategy are more efficient than the standard I controller.

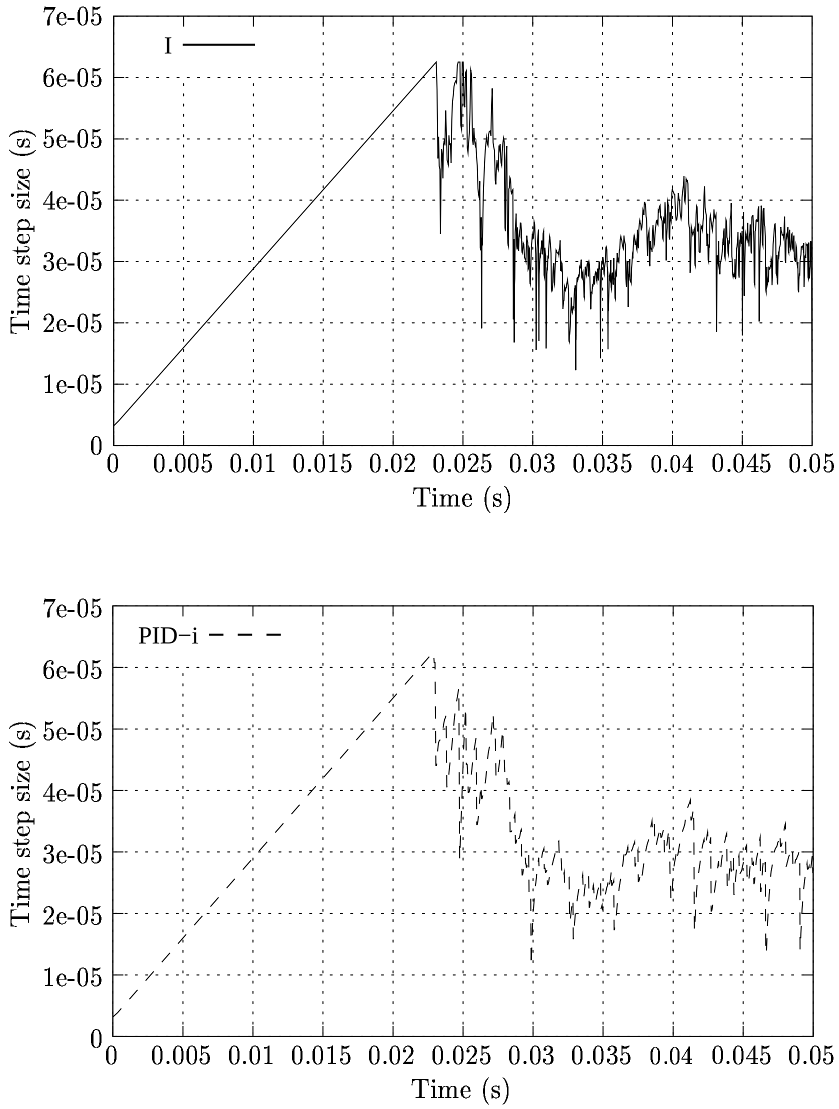


Figure 5.22: Deformable dummy impacting against an airbag. Evolution in time of the time step size. (Top: I controller. Bottom: PID-i controller).

Chapter 6

Closure

Whereas the field of error estimation and adaptivity in solid static problems has received great attention in the literature, this is not the case of deformable solid dynamics. In this dissertation, we have studied aspects of the numerical analysis and practical appliactions of error estimation to solid and structural dynamics and also we have designed algorithms for controlling the error due to the temporal discretization in dynamic analyses within the context of structural and solid mechanics.

The error estimators developed need a small computational cost and provide an useful information to ensure the accuracy of numerical results obtained in solid dynamic simulations. Some of the existing estimators do not compute upper bounds for the real error and therefore, are not reliable. Others produce good approximations to the real errors in smooth problems but do not have a good performance in non-smooth simulations. In these cases, they compute excessively large estimates and thus, the information about the error made is not very useful. These type of problems are more and more demanded in real applications and cover cases such as contacts between different bodies, impacts, abrupt external loads, etc.

With respect to the adaptive strategies for selecting the time step size, the most common used algorithms are heuristic rules. Again, these algorithms provide efficient time step sequences in smooth problems. In more complex cases, they lead to the rejection of many steps degrading their performance and increasing the computational cost. In this work, we have approached the problem of selection of the time step size from a different point of view, based on control theory.

Next, we summarized the most relevant results of this work:

i. We have deduced in Chapter 3 the conditions which an error estimator must verify to provide accurate and reliable estimates in deformable solid dynamics. Using these conditions, two existing error estimators have been analyzed from a theoretical point of view and also through several numerical simulations. The results shown by the simulations agree with theoretical conclusions.

ii. In Chapter 4, we have introduced a novel strategy for developing local error estimators for some of the most used time-stepping methods. This strategy computes an "improved" solution using a quadrature and it has been particularized to Newmark's method and to the HHT method. This class of new estimators can be employed both in linear and non-linear analyses, require a small computational cost and provide accurate estimates. Its performance has been extensively studied in six different numerical examples.

iii. In the last part (Chapter 5), adaptive strategies based on control theory have been designed. This approach allow to employ different techniques existing in the field of control theory to obtain efficient adaptive algorithms. They change automatically the time step size with the goal of solving each step with the desired accuracy, keeping at the same time the computational cost as low as possible. Using these adaptive schemes an improvement in the efficiency around 15% respect to standard algorithms is achieved.

In this work, we have focused on studying the error in time-stepping methods. As discussed in the introduction, this temporal error is only a part of the whole error. Some suggestions for future research try to cover this issue. We propose the following lines:

- Design estimators that take into account the error due to both spatial and temporal discretization. This is a complex task since, usually, the numerical methods used for each part are different.
- Build adaptive methods in space and in time. That is, they must change the time step size and also refine the spatial mesh to control the total error. With this, the problem of solid dynamics could be solved as accurate as desired.

Bibliography

M. Ainsworth & J. T. Oden [2000]. *A posteriori error estimation in finite element analysis*. John Wiley & Sons.

J. E. Akin [2005]. *Finite element analysis with error estimators*. Butterworth-Heinemann.

Ansys [2005]. *ANSYS Theory Reference*. ANSYS Inc.

U. M. Ascher & L. R. Petzold [1998]. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM.

K. J. Bathe [1996]. *Finite elements procedures*. Prentice-Hall.

T. Belytschko [1983]. An overview of semidiscretization and time integration procedures. In T. Belytschko & T. Hughes, eds., *Computational methods for transient analysis* 1–65. Elsevier Science Publishers.

T. Belytschko, W. K. Liu & B. Moran [2000]. *Nonlinear finite elements for continua and structures*. John Wiley & Sons.

D. Braess [1997]. *Finite elements. Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press.

C. K. Choi & H. J. Chung [1996]. Error estimates and adaptive time stepping for various direct time integration methods. *Computers & Structures* **60**: 923–944.

J. Chung, E.-H. Cho & K. Choi [2003]. A priori error estimator of the generalized-α method for structural dynamics. *International Journal for Numerical Methods in Engineering* **57**: 537–554.

J. Chung & G. M. Hulbert [1993]. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics* **60**: 371–375.

M. A. Crisfield [1997]. *Non-Linear finite element analysis of solids and structures. Vol. 2: Advanced topics*. Wiley.

S. Erlicher, L. Bonaventura & O. S. Bursi [2002]. The analysis of the generalized-α method for non-linear dynamic problems. *Computational Mechanics* **28**: 83–104.

J. García de Jalón & E. Bayo [1993]. *Kinematic and dynamic simulation of multibody systems: The real-time challenge*. Mechanical Engineering Series. Springer-Verlag.

C. W. Gear [1971]. *Numerical initial value problems in ordinary differential equations*. Series in automatic computation. Prentice-Hall.

M. Gèradin & A. Cardona [1997]. *Flexible multibody dynamics: a finite element approach* cap. 11, 277–293. John Wiley & Sons 1 edón.

K. Gustafsson [1991]. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. *ACM Transactions on Mathematical Software* **17**, nº 4: 533–554.

K. Gustafsson [1994]. Control theoretic techniques for stepsize selection in implicit Runge-Kutta methods. *ACM Transactions on Mathematical Software* **20**, nº 4: 496–517.

K. Gustafsson, M. Lundh & G. Soderlind [1988]. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT* **28**, nº 2: 270–287.

E. Hairer, S. P. Norsett & G. Wanner [1987]. *Solving ordinary differential equation I: nonstiff problems*. Springer Series in Computational Mathematics. Springer-Verlag.

E. Hairer & G. Wanner [1991]. *Solving ordinary differential equation II: stiff and differential-algebraic problems*. Springer Series in Computational Mathematics. Springer-Verlag.

Hibbit, Karlsson & Sorensen [2004]. *Abaqus Theory Manual, version 6.5*. ABAQUS.

H. M. Hilber, T. J. R. Hughes & R. L. Taylor [1977]. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics* **5**: 283–292.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker & C. S. Woodward [2005]. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software* **31**: 363–396.

J. C. HOUBOLT [1950]. A recurrence matrix solution for the dynamic response of elastic aircraft. *Journal of Aeronautical Sciences* **17**: 540–550.

T. J. R. HUGHES [1983]. Analysis of transient algorithms with particular reference to stability behavior. In T. Belytschko & T. Hughes, eds., *Computational methods for transient analysis* 67–155. Elsevier Science Publishers.

T. J. R. HUGHES [2000]. *The finite element method: linear static and dynamic finite element analysis*. Dover.

G. M. HULBERT [1989]. *Space-time finite element methods for second-order hyperbolic equations*. PhD. thesis, Stanford University.

G. M. HULBERT & T. J. R. HUGHES [1987]. An error analysis of truncated starting conditions in step-by-step time integration: Consequences for structural dynamics. *Earthquake Engineering and Structural Dynamics* **15**: 901–910.

G. M. HULBERT & I. JANG [1995]. Automatic time step control algorithms for structural dynamics. *Computer Methods in Applied Mechanics and Engineering* **126**: 155–178.

D. KUHL & M. A. CRISFIELD [1999]. Energy-conserving and decayimg algorithms in non-linear structural dynamics. *International Journal for Numerical Methods in Engineering* **45**: 569–599.

L. M. LACOMA & I. ROMERO [2006a]. Error estimation for the HHT method in non-linear solid dynamics. *Submitted to Computers & Structures*.

L. M. LACOMA & I. ROMERO [2006b]. Estimación de error en dinámica de sólidos deformables. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* **22**: 45–62.

P. LADEVÈZE, G. PUEL, A. DERAEMAEKER & T. ROMEUF [2006a]. Validation of structural dynamics models containig uncertainties. *Computer Methods in Applied Mechanics and Engineering* **195**: 373–393.

P. LADEVÈZE, G. PUEL & T. ROMEUF [2006b]. Lack of knowledge in structural model validation. *Computer Methods in Applied Mechanics and Engineering* **In press**.

J. D. LAMBERT [1991]. *Numerical methods for ordinary differential systems*. John Wiley & Sons.

X. D. LI & N. E. WIBERG [1998]. Implementation and adaptivity of a space-time finite element methods for structural dynamics. *Computer Methods in Applied Mechanics and Engineering* **156**: 211–229.

N. M. Newmark [1959]. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division, ASCE* 67–94.

L. Noels, L. Stainier, J. P. Ponthot & J. Bonini [2003]. Automatic time stepping algorithms for implicit numerical simulations of non-linear dynamics. *Advances in Engineering Software* **33**: 589–603.

C. H. Phillips & H. T. Nagle [1998]. *Digital control system analysis and design*. Prentice-Hall 3 edón.

Plaxis [2005]. *Dynamics Manual for v8*. PLAXIS BV.

W. H. Press, S. A. Teukolsky, W. T. Vetterling & B. P. Flannery [1992]. *Numerical recipes in C. The art of scientific computation*. Cambridge University Press 2 edón.

I. Romero [2002]. On the stability and convergence of fully discrete solutions in linear elastodynamics. *Computer Methods in Applied Mechanics and Engineering* **191**: 3857–3882.

I. Romero [2004]. Stability analysis of linear multistep methods for classical elastodynamics. *Computer Methods in Applied Mechanics and Engineering* **193**: 2169–2189.

I. Romero & L. M. Lacoma [2004]. Estimación a posteriori de error en simulaciones de elastodinámica lineal. In *Congreso de Métodos Computacionales en Ingeniería*. APMTAC.

I. Romero & L. M. Lacoma [2006a]. Analysis and improved methods for the error estimation of numerical solutions in solid and structural dynamics. *Submitted to Springer Verlag, Advances in Computational Multibody Dynamics*.

I. Romero & L. M. Lacoma [2006b]. Error estimation for the semidiscrete equations of solid and structural mechanics. *Computer Methods in Applied Mechanics and Engineering* **195**: 2674–2696.

I. Romero & L. M. Lacoma [2006c]. A methodology for the formulation of error estimators for time integration in solid and structural dynamics. *International Journal for Numerical Methods in Engineering* **66**: 635–660.

A. Schleupen & E. Ramm [2000]. Local and global error estimations in linear structural dynamics. *Computers & Structures* **76**: 741–756.

L. F. Shampine [To appear]. Error estimation and control for odes. *Journal of Scientific Computing*.

G. Söderlind [2002]. Automatic control and adaptive time-stepping. *Numerical Algorithms* **31**: 281–310.

G. Söderlind [2003]. Digital filters in adaptive time-stepping. *ACM Transactions on Mathematical Software* **29**: 1–26.

G. Söderlind [2006]. Time-step selection algorithms: Adaptivity, control and signal processing. *Applied Numerical Mathematics* **In press**.

G. Söderlind & L. Wang [2006]. Adaptive time-stepping and computational stability. *Journal of Computational and Applied Mathematics* **185**: 225–243.

A. M. Stuart & A. R. Humphries [1996]. *Dynamical systems and numerical analysis*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

K. K. Tamma, X. Zhou & D. Sha [2000]. The time dimension: A theory towards the evolution, classification, characterization and design of computational algorithms for transient/dynamic applications. *Archives of Computational Methods in Engineering* **7**: 67–286.

N. Tarnow[1993]. *Energy and momentum conserving algorithms for Hamiltonian systems in the nonlinear dynamics of solids*. PhD Thesis, Sudam Report No. 93-4, Department of Mechanical Engineering, Stanford University, California.

A. M. P. Valli, G. F. Carey & A. L. G. A. Coutinho [2002]. Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Communications in Numerical Methods in Engineering* **18**: 131–139.

E. W. Weisstein [2006]. *"Lobatto Quadrature"*. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/LobattoQuadrature.html.

N. E. Wiberg & X. D. Li [1993]. A post-processing technique and a posteriori error estimate for the newmark method in dynamic analysis. *Earthquake Engineering and Structural Dynamics* **22**: 465–489.

E. Wilson [1968]. A computer program for the dynamic stress analysis of underground structures. EERC Report No. 68-1. Division of Structural Engineering and Structural Mechanics, University of California, Berkeley.

L. F. Zeng, N. E. Wiberg, X. D. Li & Y. M. Xie [1992]. A posteriori local error estimation and adaptive time-stepping for newmark integration in dynamic analysis. *Earthquake Engineering and Structural Dynamics* **21**: 555–571.

O. C. Zienkiewicz & Y. M. Xie [1991]. A simple error estimator and adaptive time stepping procedure for dynamic analysis. *Earthquake Engineering and Structural Dynamics* **20**: 871–887.